

Technion – Israel Institute of Technology
The William Davidson Faculty of Industrial Engineering & Management
Haifa Israel



Georgia Institute of Technology
H. Milton Stewart School of Industrial & Systems Engineering
Atlanta Georgia USA

LECTURE NOTES

OPTIMIZATION III

CONVEX ANALYSIS

NONLINEAR PROGRAMMING THEORY

NONLINEAR PROGRAMMING ALGORITHMS

ISYE 6663

Aharon Ben-Tal[†] & Arkadi Nemirovski[*]

[†]The William Davidson Faculty of Industrial Engineering & Management,
Technion – Israel Institute of Technology
[*]H. Milton Stewart School of Industrial & Systems Engineering,
Georgia Institute of Technology



Spring Semester 2022

Aim: Introduction to the Theory of Nonlinear Programming and algorithms of Continuous Optimization.

Duration: 14 weeks, 3 hours per week

Prerequisites: elementary Linear Algebra (vectors, matrices, Euclidean spaces); basic knowledge of Calculus (including gradients and Hessians of multivariate functions). These prerequisites are summarized in Appendix.

Contents:

Part I. Elements of Convex Analysis and Optimality Conditions

7 weeks

1-2. Convex sets (definitions, basic properties, Caratheodory-Radon-Helley theorems)

3-4. The Separation Theorem for convex sets (Farkas Lemma, Separation, Theorem on Alternative, Extreme points, Krein-Milman Theorem in $\mathbf{R}^n$, structure of polyhedral sets, theory of Linear Programming)

5. Convex functions (definition, differential characterizations, operations preserving convexity)

6. Mathematical Programming programs and Lagrange duality in Convex Programming (Convex Programming Duality Theorem with applications to linearly constrained convex Quadratic Programming)

7. Optimality conditions in unconstrained and constrained optimization (Fermat rule; Karush-Kuhn-Tucker first order optimality condition for the regular case; necessary/sufficient second order optimality conditions for unconstrained case; second order sufficient optimality conditions)

Part II: Algorithms

7 weeks

8. Univariate unconstrained minimization (Bisection; Curve Fitting; Armijo-terminated inexact line search)

9. Multivariate unconstrained minimization: Gradient Descent and Newton methods

10. Around the Newton Method (variable metric methods, conjugate gradients, quasi-Newton algorithms)

11. Polynomial solvability of Convex Programming

12. Constrained minimization: active sets and penalty/barrier approaches

13. Constrained minimization: augmented Lagrangians

14. Constrained minimization: Sequential Quadratic Programming

Contents

1 Convex sets in $\mathbf{R}^n$ **11**

1.1 Definition and basic properties . . . 11
1.1.1 A convex set . . . 11
1.1.2 Examples of convex sets . . . 11
1.1.2.1 Affine subspaces and polyhedral sets . . . 11
1.1.2.2 Unit balls of norms . . . 12
1.1.2.3 Ellipsoids . . . 14
1.1.2.4 Neighbourhood of a convex set . . . 14
1.1.3 Inner description of convex sets: Convex combinations and convex hull . . . 14
1.1.3.1 Convex combinations . . . 14
1.1.3.2 Convex hull . . . 15
1.1.3.3 Simplex . . . 15
1.1.4 Cones . . . 16
1.1.5 Calculus of convex sets . . . 17
1.1.6 Topological properties of convex sets . . . 17
1.1.6.1 The closure . . . 17
1.1.6.2 The interior . . . 18
1.1.6.3 The relative interior . . . 19
1.1.6.4 Nice topological properties of convex sets . . . 20
1.2 Main theorems on convex sets . . . 22
1.2.1 Caratheodory Theorem . . . 22
1.2.2 Radon Theorem . . . 23
1.2.3 Helley Theorem . . . 23
1.2.4 Polyhedral representations and Fourier-Motzkin Elimination . . . 24
1.2.4.1 Polyhedral representations . . . 24
1.2.4.2 Every polyhedrally representable set is polyhedral! (Fourier-Motzkin elimination) . . . 25
1.2.4.3 Some applications . . . 26
1.2.4.4 Calculus of polyhedral representations . . . 27
1.2.5 General Theorem on Alternative and Linear Programming Duality . . . 28
1.2.5.1 Homogeneous Farkas Lemma . . . 28
1.2.5.2 General Theorem on Alternative . . . 31
1.2.5.3 Application: Linear Programming Duality . . . 34
1.2.6 Separation Theorem . . . 38
1.2.6.1 Separation: definition . . . 38
1.2.6.2 Separation Theorem . . . 40

1.2.6.3 Supporting hyperplanes . . . 44
1.2.7 Polar of a convex set and Dubovitski-Milutin Lemma . . . 44
1.2.7.1 Polar of a convex set . . . 44
1.2.7.2 Dual cone . . . 45
1.2.7.3 Dubovitski-Milutin Lemma . . . 46
1.2.8 Extreme points and Krein-Milman Theorem . . . 47
1.2.8.1 Extreme points: definition . . . 47
1.2.8.2 Krein-Milman Theorem . . . 48
1.2.8.3 Example: Extreme points of a polyhedral set. . . . 50
1.2.9 Structure of polyhedral sets . . . 52
1.2.9.1 Main result . . . 52
1.2.9.2 Theory of Linear Programming . . . 54
1.2.9.3 Structure of a polyhedral set: proofs . . . 56

2 Convex functions **59**
2.1 Convex functions: first acquaintance . . . 59
2.1.1 Definition and Examples . . . 59
2.1.2 Elementary properties of convex functions . . . 60
2.1.2.1 Jensen's inequality . . . 60
2.1.2.2 Convexity of level sets of a convex function . . . 61
2.1.3 What is the value of a convex function outside its domain? . . . 61
2.2 How to detect convexity . . . 62
2.2.1 Operations preserving convexity of functions . . . 62
2.2.2 Differential criteria of convexity . . . 64
2.3 Gradient inequality . . . 67
2.4 Boundedness and Lipschitz continuity of a convex function . . . 68
2.5 Maxima and minima of convex functions . . . 71
2.6 Subgradients and Legendre transformation . . . 76
2.6.1 Proper functions and their representation . . . 76
2.6.2 Subgradients . . . 82
2.6.3 Legendre transformation . . . 83

3 Convex Programming, Lagrange Duality, Saddle Points **87**
3.1 Mathematical Programming Program . . . 87
3.2 Convex Programming program and Lagrange Duality Theorem . . . 88
3.2.1 Convex Theorem on Alternative . . . 88
3.2.1.1 Cone constrained case . . . 90
3.2.2 Lagrange Function and Lagrange Duality . . . 95
3.2.2.1 Lagrange function . . . 95
3.2.2.2 Convex Programming Duality Theorem . . . 96
3.2.2.3 Dual program . . . 97
3.2.2.4 Cone constrained forms of Lagrange Function, Lagrange Duality, and Convex Programming Duality Theorem . . . 97
3.2.2.5 Conic Programming and Conic Duality Theorem . . . 98
3.2.3 Optimality Conditions in Convex Programming . . . 99
3.2.3.1 Saddle point form of optimality conditions . . . 99
3.2.3.2 Karush-Kuhn-Tucker form of optimality conditions . . . 101

3.2.3.3 Optimality conditions in Conic Programming . . . 102
3.3 Duality in Linear and Convex Quadratic Programming . . . 103
3.3.1 Linear Programming Duality . . . 103
3.3.2 Quadratic Programming Duality . . . 104
3.4 Saddle Points . . . 106
3.4.1 Definition and Game Theory interpretation . . . 106
3.4.2 Existence of Saddle Points . . . 108

4 Optimality Conditions 113
4.1 First Order Optimality Conditions . . . 115
4.2 Second Order Optimality Conditions . . . 121
4.2.1 Nondegenerate solutions and Sensitivity Analysis . . . 130
4.3 Concluding Remarks . . . 135

5 Optimization Methods: Introduction 141
5.1 Preliminaries on Optimization Methods . . . 142
5.1.1 Classification of Nonlinear Optimization Problems and Methods . . . 142
5.1.2 Iterative nature of optimization methods . . . 142
5.1.3 Convergence of Optimization Methods . . . 143
5.1.3.1 Rates of convergence . . . 143
5.1.4 Global and Local solutions . . . 145
5.2 Line Search . . . 146
5.2.1 Zero-Order Line Search . . . 147
5.2.1.1 Fibonacci search . . . 148
5.2.1.2 Golden search . . . 150
5.2.2 Bisection . . . 151
5.2.3 Curve fitting . . . 153
5.2.3.1 Newton's method . . . 153
5.2.3.2 Regula Falsi (False Position) method . . . 155
5.2.3.3 Cubic fit . . . 157
5.2.3.4 Safeguarded curve fitting . . . 157
5.2.4 Inexact Line Search . . . 158
5.2.4.1 Armijo's rule . . . 158
5.2.4.2 Goldstein test . . . 159

6 Gradient Descent and Newton Methods 161
6.1 Gradient Descent . . . 161
6.1.1 The idea . . . 161
6.1.2 Standard implementations . . . 162
6.1.3 Convergence of the Gradient Descent . . . 163
6.1.3.1 General Convergence Theorem . . . 163
6.1.3.2 Limiting points of Gradient Descent . . . 165
6.1.4 Rates of convergence . . . 165
6.1.4.1 Rate of global convergence: general $C^{1,1}$ case . . . 165
6.1.4.2 Rate of global convergence: convex $C^{1,1}$ case . . . 168
6.1.4.3 Rate of convergence in the quadratic case . . . 173
6.1.5 Conclusions . . . 176

6.2 Basic Newton's Method . . . 177
6.2.1 The Method . . . 178
6.2.2 Incorporating line search . . . 179
6.2.3 The Newton Method: how good it is? . . . 180
6.2.4 Newton Method and Self-Concordant Functions . . . 181
6.2.4.1 Preliminaries . . . 181
6.2.4.2 Self-concordance . . . 182
6.2.4.3 Self-concordant functions and the Newton method . . . 183
6.2.4.4 Self-concordant functions: applications . . . 185

7 Around the Newton Method 189
7.1 Newton Method with Cubic Regularization . . . 190
7.1.1 Implementing the algorithm . . . 192
7.2 Modified Newton methods . . . 193
7.2.1 Variable Metric Methods . . . 193
7.2.2 Global convergence of a Variable Metric method . . . 195
7.2.3 Implementations of the Modified Newton method . . . 196
7.2.3.1 Modifications based on Spectral Decomposition . . . 196
7.2.3.2 Levenberg-Marquardt Modification . . . 197
7.2.3.3 Choleski Factorization . . . 198
7.3 Conjugate Gradient Methods . . . 198
7.3.1 Conjugate Gradient Method: Quadratic Case . . . 199
7.3.1.1 CG: Initial description . . . 200
7.3.1.2 Iterative representation of the Conjugate Gradient method . . . 200
7.3.1.3 CG and Three-Diagonal representation of a Symmetric matrix . . . 204
7.3.1.4 Rate of convergence of the Conjugate Gradient method . . . 204
7.3.1.5 Conjugate Gradient algorithm for quadratic minimization: advantages and disadvantages . . . 208
7.3.2 Extensions to non-quadratic problems . . . 209
7.3.3 Global and local convergence of Conjugate Gradient methods in non-quadratic case . . . 210
7.4 Quasi-Newton Methods . . . 211
7.4.1 The idea . . . 212
7.4.2 The Generic Quasi-Newton Scheme . . . 212
7.4.3 Implementations . . . 214
7.4.3.1 Davidon-Fletcher-Powell method . . . 214
7.4.3.2 The Broyden family . . . 215
7.4.4 Convergence of Quasi-Newton methods . . . 216
7.4.4.1 Global convergence . . . 216
7.4.4.2 Local convergence . . . 217
7.4.4.3 Appendix: derivation of the BFGS updating formula . . . 218

8 Convex Programming 219
8.1 Preliminaries . . . 219
8.1.1 Subgradients of convex functions . . . 220
8.1.2 Separating planes . . . 220
8.2 The Ellipsoid Method . . . 221

8.2.1 The idea . . . 221
8.2.2 The Center-of-Gravity method . . . 222
8.2.3 From Center-of-Gravity to the Ellipsoid method . . . 223
8.2.4 The Algorithm . . . 225
8.2.4.1 How to represent an ellipsoid . . . 225
8.2.4.2 The Ellipsoid algorithm . . . 226
8.2.5 The Ellipsoid algorithm: rate of convergence . . . 226
8.2.6 Ellipsoid method for problems with functional constraints . . . 228
8.3 Ellipsoid method and Complexity of Convex Programming . . . 230
8.3.1 Complexity: what is it? . . . 230
8.3.2 Computational Tractability = Polynomial Solvability . . . 232
8.3.3 R-Polynomial Solvability of Convex Programming . . . 233
8.4 Polynomial solvability of Linear Programming . . . 234
8.4.1 Polynomial Solvability of Linear Programming over Rationals . . . 234
8.4.1.1 Some History . . . 234
8.4.2 Khachiyan's Theorem . . . 235
8.4.2.1 Step 1: from Optimization to Feasibility . . . 235
8.4.2.2 Step 2: from Feasibility to Solvability . . . 236
8.4.2.3 From Solvability back to Feasibility . . . 238
8.4.3 More History . . . 239

9 Active Set and Penalty/Barrier Methods 241
9.1 Primal methods . . . 242
9.1.1 Methods of Feasible Directions . . . 242
9.1.2 Active Set Methods . . . 243
9.1.2.1 Active Set scheme: the idea . . . 243
9.1.2.2 Active Set scheme: implementation . . . 244
9.1.2.3 Active Set Scheme: convergence . . . 247
9.1.2.4 Standard applications: Linear and Quadratic Programming . . . 248
9.2 Penalty and Barrier Methods . . . 249
9.2.1 The idea . . . 250
9.2.2 Penalty methods . . . 252
9.2.2.1 Convergence . . . 252
9.2.2.2 Properties of the path $x^*(\rho)$ . . . 259
9.2.3 Barrier methods . . . 262
9.2.3.1 Self-concordant barriers and path-following scheme . . . 263
9.2.3.2 Path-following scheme . . . 264
9.2.3.3 Applications . . . 268
9.2.3.4 Concluding remarks . . . 269

10 Augmented Lagrangians 271
10.1 Main ingredients . . . 271
10.1.1 Local Lagrange Duality . . . 271
10.1.2 "Penalty cocknification" . . . 273
10.2 Putting things together: Augmented Lagrangian Scheme . . . 275
10.2.1 Solving auxiliary primal problems (P_λ^ρ) . . . 276
10.2.2 Solving the dual problem . . . 276

10.2.2.1 Dual rate of convergence . . . 277
10.2.3 Adjusting penalty parameter . . . 279
10.3 Incorporating Inequality Constraints . . . 279
10.4 Convex case: Augmented Lagrangians; Decomposition . . . 280
10.4.1 Augmented Lagrangians . . . 282
10.4.2 Lagrange Duality and Decomposition . . . 283

11 Sequential Quadratic Programming 287
11.1 SQP methods: Equality Constrained case . . . 287
11.1.1 Newton method for systems of equations . . . 287
11.1.1.1 The method . . . 287
11.1.1.2 Local quadratic convergence . . . 288
11.1.2 Solving (KKT) by the Newton method . . . 289
11.1.2.1 Nonsingularity of KKT points . . . 289
11.1.2.2 Structure and interpretation of the Newton displacement . . . 290
11.2 The case of general constrained problems . . . 292
11.2.1 Basic SQP scheme . . . 292
11.2.2 Quasi-Newton Approximations . . . 293
11.3 Linesearch, Merit functions, global convergence . . . 294
11.3.1 l_1 merit function . . . 295
11.3.2 SQP Algorithm with Merit Function . . . 297
11.3.3 Concluding remarks . . . 298

A Prerequisites from Linear Algebra and Analysis 301
A.1 Space $\mathbf{R}^n$: algebraic structure . . . 301
A.1.1 A point in $\mathbf{R}^n$ . . . 301
A.1.2 Linear operations . . . 301
A.1.3 Linear subspaces . . . 302
A.1.4 Linear independence, bases, dimensions . . . 303
A.1.5 Linear mappings and matrices . . . 304
A.2 Space $\mathbf{R}^n$: Euclidean structure . . . 306
A.2.1 Euclidean structure . . . 306
A.2.2 Inner product representation of linear forms on $\mathbf{R}^n$ . . . 307
A.2.3 Orthogonal complement . . . 307
A.2.4 Orthonormal bases . . . 308
A.3 Affine subspaces in $\mathbf{R}^n$ . . . 310
A.3.1 Affine subspaces and affine hulls . . . 310
A.3.2 Intersections of affine subspaces, affine combinations and affine hulls . . . 311
A.3.3 Affinely spanning sets, affinely independent sets, affine dimension . . . 312
A.3.4 Dual description of linear subspaces and affine subspaces . . . 314
A.3.4.1 Affine subspaces and systems of linear equations . . . 315
A.3.5 Structure of the simplest affine subspaces . . . 316
A.4 Space $\mathbf{R}^n$: metric structure and topology . . . 317
A.4.1 Euclidean norm and distances . . . 317
A.4.2 Convergence . . . 318
A.4.3 Closed and open sets . . . 319
A.4.4 Local compactness of $\mathbf{R}^n$ . . . 320

A.5 Continuous functions on $\mathbf{R}^n$ . . . 320
A.5.1 Continuity of a function . . . 320
A.5.2 Elementary continuity-preserving operations . . . 321
A.5.3 Basic properties of continuous functions on $\mathbf{R}^n$ . . . 321
A.6 Differentiable functions on $\mathbf{R}^n$ . . . 322
A.6.1 The derivative . . . 322
A.6.2 Derivative and directional derivatives . . . 324
A.6.3 Representations of the derivative . . . 325
A.6.4 Existence of the derivative . . . 326
A.6.5 Calculus of derivatives . . . 327
A.6.6 Computing the derivative . . . 327
A.6.7 Higher order derivatives . . . 329
A.6.8 Calculus of C^k mappings . . . 331
A.6.9 Examples of higher-order derivatives . . . 332
A.6.10 Taylor expansion . . . 333
A.7 Symmetric matrices . . . 334
A.7.1 Spaces of matrices . . . 334
A.7.2 Main facts on symmetric matrices . . . 335
A.7.3 Variational characterization of eigenvalues . . . 336
A.7.3.1 Corollaries of the VCE . . . 338
A.7.4 Positive semidefinite matrices and the semidefinite cone . . . 339

Lecture 1

Convex sets in $\mathbf{R}^n$

1.1 Definition and basic properties

1.1.1 A convex set

In the school geometry a figure is called convex if it contains, along with every pair of its points x, y, also the entire segment $[x, y]$ linking the points. This is exactly the definition of a convex set in the multidimensional case; all we need is to say what does it mean "the segment $[x, y]$ linking the points $x, y \in \mathbf{R}^n$". This is said by the following

Definition 1.1.1 [Convex set]
1) *Let x, y be two points in $\mathbf{R}^n$. The set*

$$[x, y] = \{z = \lambda x + (1 - \lambda)y : 0 \leq \lambda \leq 1\}$$

is called a segment with the endpoints x, y.
2) *A subset M of $\mathbf{R}^n$ is called convex, if it contains, along with every pair of its points x, y, also the entire segment $[x, y]$:*

$$x, y \in M,\ 0 \leq \lambda \leq 1 \Rightarrow \lambda x + (1 - \lambda)y \in M.$$

Note that by this definition an empty set is convex (by convention, or better to say, by the exact sense of the definition: for the empty set, you cannot present a counterexample to show that it is not convex).

1.1.2 Examples of convex sets

1.1.2.1 Affine subspaces and polyhedral sets

Example 1.1.1 *A linear/affine subspace of $\mathbf{R}^n$ is convex.*

Convexity of affine subspaces immediately follows from the possibility to represent these sets as solution sets of systems of linear equations (Proposition A.3.7), due to the following simple and important fact:

Proposition 1.1.1 *The solution set of an arbitrary (possibly, infinite) system*

$$a_\alpha^T x \leq b_\alpha,\ \alpha \in \mathcal{A} \qquad (!)$$

of nonstrict linear inequalities with n unknowns x – the set

$$S = \{x \in \mathbf{R}^n : a_\alpha^T x \le b_\alpha,\ \alpha \in \mathcal{A}\}$$

is convex.

In particular, the solution set of a finite system

$$Ax \le b$$

of m nonstrict inequalities with n variables (A is $m \times n$ matrix) is convex; a set of this latter type is called polyhedral.

Exercise 1.1 *Prove Proposition 1.1.1.*

Remark 1.1.1 *Note that every set given by Proposition 1.1.1 is not only convex, but also closed (why?). In fact, from Separation Theorem (Theorem 1.2.9 below) it follows that*

> *Every closed convex set in $\mathbf{R}^n$ is the solution set of a (perhaps, infinite) system of nonstrict linear inequalities.*

Remark 1.1.2 *Note that replacing some of the nonstrict linear inequalities $a_\alpha^T x \le b_\alpha$ in (!) with their strict versions $a_\alpha^T x < b_\alpha$, we get a system with the solution set which still is convex (why?), but now not necessary is closed.*

1.1.2.2 Unit balls of norms

Let $\|\cdot\|$ be a norm on $\mathbf{R}^n$ i.e., a real-valued function on $\mathbf{R}^n$ satisfying the three characteristic properties of a norm (Section A.4.1), specifically:

A. [positivity] $\| x \| \ge 0$ for all $x \in \mathbf{R}^n$; $\| x \| = 0$ is and only if $x = 0$;

B. [homogeneity] For $x \in \mathbf{R}^n$ and $\lambda \in \mathbf{R}$, one has

$$\| \lambda x \| = |\lambda| \, \| x \|;$$

C. [triangle inequality] For all $x, y \in \mathbf{R}^n$ one has

$$\| x + y \| \le \| x \| + \| y \| .$$

Example 1.1.2 *The unit ball of the norm $\|\cdot\|$ – the set*

$$\{x \in E : \| x \| \le 1\},$$

same as every other $\|\cdot\|$-ball

$$\{x : \| x - a \| \le r\}$$

($a \in \mathbf{R}^n$ and $r \ge 0$ are fixed) is convex.

In particular, Euclidean balls ($\|\cdot\|$-balls associated with the standard Euclidean norm $\| x \|_2 = \sqrt{x^T x}$) are convex.

The standard examples of norms on $\mathbf{R}^n$ are the ℓ_p-norms

$$\| x \|_p = \begin{cases} \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, & 1 \le p < \infty \\ \max_{1 \le i \le n} |x_i|, & p = \infty \end{cases}.$$

These indeed are norms (which is not clear in advance). When $p = 2$, we get the usual Euclidean norm; of course, you know how the Euclidean ball looks. When $p = 1$, we get

$$\| x \|_1 = \sum_{i=1}^n |x_i|,$$

and the unit ball is the *hyperoctahedron*

$$V = \{x \in \mathbf{R}^n : \sum_{i=1}^n |x_i| \le 1\}$$

When $p = \infty$, we get

$$\| x \|_\infty = \max_{1 \le i \le n} |x_i|,$$

and the unit ball is the *hypercube*

$$V = \{x \in \mathbf{R}^n : -1 \le x_i \le 1, \, 1 \le i \le n\}.$$

Exercise 1.2 *Prove that unit balls of norms on $\mathbf{R}^n$ are exactly the same as convex sets V in $\mathbf{R}^n$ satisfying the following three properties:*

1. *V is symmetric w.r.t. the origin: $x \in V \Rightarrow -x \in V$;*
2. *V is bounded and closed;*
3. *V contains a neighbourhood of the origin.*

A set V satisfying the outlined properties is the unit ball of the norm

$$\| x \| = \inf \left\{ t \ge 0 : t^{-1} x \in V \right\}.$$

Hint: You could find useful to verify and to exploit the following facts:

1. A norm $\| \cdot \|$ on $\mathbf{R}^n$ is Lipschitz continuous with respect to the standard Euclidean distance: there exists $C_{\| \cdot \|} < \infty$ such that $| \, \| x \| - \| y \| \, | \le C_{\| \cdot \|} \| x - y \|_2$ for all x, y
2. Vice versa, the Euclidean norm is Lipschitz continuous with respect to a given norm $\| \cdot \|$: there exists $c_{\| \cdot \|} < \infty$ such that $| \, \| x \|_2 - \| y \|_2 \, | \le c_{\| \cdot \|} \| x - y \|$ for all x, y

1.1.2.3 Ellipsoids

Example 1.1.3 [Ellipsoid] *Let Q be a $n \times n$ matrix which is symmetric ($Q = Q^T$) and positive definite ($x^T Q x \geq 0$, with $\geq$ being $=$ if and only if $x = 0$). Then, for every nonnegative r, the Q-ellipsoid of radius r centered at a – the set*

$$\{x : (x-a)^T Q (x-a) \leq r^2\}$$

is convex.

To see that an ellipsoid $\{x : (x-a)^T Q(x-a) \leq r^2\}$ is convex, note that since Q is positive definite, the matrix $Q^{1/2}$ is well-defined and positive definite. Now, if $\|\cdot\|$ is a norm on $\mathbf{R}^n$ and P is a nonsingular $n \times n$ matrix, the function $\|Px\|$ is a norm along with $\|\cdot\|$ (why?). Thus, the function $\|x\|_Q \equiv \sqrt{x^T Q x} = \|Q^{1/2} x\|_2$ is a norm along with $\|\cdot\|_2$, and the ellipsoid in question clearly is just $\|\cdot\|_Q$-ball of radius r centered at a.

1.1.2.4 Neighbourhood of a convex set

Example 1.1.4 *Let M be a convex set in $\mathbf{R}^n$, and let $\epsilon > 0$. Then, for every norm $\|\cdot\|$ on $\mathbf{R}^n$, the ϵ-neighbourhood of M, i.e., the set*

$$M_\epsilon = \{y \in \mathbf{R}^n : \mathrm{dist}_{\|\cdot\|}(y, M) \equiv \inf_{x \in M} \|y - x\| \leq \epsilon\}$$

is convex.

Exercise 1.3 *Justify the statement of Example 1.1.4.*

1.1.3 Inner description of convex sets: Convex combinations and convex hull

1.1.3.1 Convex combinations

Recall the notion of *linear combination* y of vectors $y_1, ..., y_m$ – this is a vector represented as

$$y = \sum_{i=1}^m \lambda_i y_i,$$

where λ_i are real coefficients. Specifying this definition, we have come to the notion of an *affine combination* - this is a linear combination with the sum of coefficients equal to one. The last notion in this genre is the one of *convex combination*.

Definition 1.1.2 *A convex combination of vectors $y_1, ..., y_m$ is their affine combination with nonnegative coefficients, or, which is the same, a linear combination*

$$y = \sum_{i=1}^m \lambda_i y_i$$

with nonnegative coefficients with unit sum:

$$\lambda_i \geq 0, \ \sum_{i=1}^m \lambda_i = 1.$$

The following statement resembles those in Corollary A.3.2:

Proposition 1.1.2 *A set M in $\mathbf{R}^n$ is convex if and only if it is closed with respect to taking all convex combinations of its elements, i.e., if and only if every convex combination of vectors from M again is a vector from M.*

Exercise 1.4 *Prove Proposition 1.1.2.*
Hint: Assuming $\lambda_1, ..., \lambda_m > 0$, one has

$$\sum_{i=1}^m \lambda_i y_i = \lambda_1 y_1 + (\lambda_2 + \lambda_3 + ... + \lambda_m) \sum_{i=2}^m \mu_i y_i, \quad \mu_i = \frac{\lambda_i}{\lambda_2 + \lambda_3 + ... + \lambda_m}.$$

1.1.3.2 Convex hull

Same as the property to be linear/affine subspace, the property to be convex is preserved by taking intersections (why?):

Proposition 1.1.3 *Let $\{M_\alpha\}_\alpha$ be an arbitrary family of convex subsets of $\mathbf{R}^n$. Then the intersection*

$$M = \cap_\alpha M_\alpha$$

is convex.

As an immediate consequence, we come to the notion of *convex hull* $\mathrm{Conv}(M)$ of a nonempty subset in $\mathbf{R}^n$ (cf. the notions of linear/affine hull):

Corollary 1.1.1 [Convex hull]
Let M be a nonempty subset in $\mathbf{R}^n$. Then among all convex sets containing M (these sets do exist, e.g., $\mathbf{R}^n$ itself) there exists the smallest one, namely, the intersection of all convex sets containing M. This set is called the convex hull of M [notation: $\mathrm{Conv}(M)$].

The linear span of M is the set of all linear combinations of vectors from M, the affine hull is the set of all affine combinations of vectors from M. As you guess,

Proposition 1.1.4 [Convex hull via convex combinations] *For a nonempty $M \subset \mathbf{R}^n$:*

$$\mathrm{Conv}(M) = \{\text{the set of all convex combinations of vectors from } M\}.$$

Exercise 1.5 *Prove Proposition 1.1.4.*

1.1.3.3 Simplex

The convex hull of $m+1$ affinely independent points $y_0, ..., y_m$ (Section A.3.3) is called *m-dimensional simplex with the vertices $y_0, .., y_m$*. By results of Section A.3.3, every point x of an m-dimensional simplex with vertices $y_0, ..., y_m$ admits exactly one representation as a convex combination of the vertices; the corresponding coefficients form the unique solution to the system of linear equations

$$\sum_{i=0}^m \lambda_i x_i = x, \sum_{i=0}^m \lambda_i = 1.$$

This system is solvable if and only if $x \in M = \mathrm{Aff}(\{y_0, .., y_m\})$, and the components of the solution (the barycentric coordinates of x) are affine functions of $x \in \mathrm{Aff}(M)$; the simplex itself is comprised of points from M with nonnegative barycentric coordinates.

1.1.4 Cones

A nonempty subset M of $\mathbf{R}^n$ is called *conic*, if it contains, along with every point $x \in M$, the entire ray $\mathbf{R}x = \{tx : t \geq 0\}$ spanned by the point:

$$x \in M \Rightarrow tx \in M \ \forall t \geq 0.$$

A convex conic set is called *a cone*.

Proposition 1.1.5 *A nonempty subset M of $\mathbf{R}^n$ is a cone if and only if it possesses the following pair of properties:*

- *is conic: $x \in M, t \geq 0 \Rightarrow tx \in M$;*
- *contains sums of its elements: $x, y \in M \Rightarrow x + y \in M$.*

Exercise 1.6 *Prove Proposition 1.1.5.*

As an immediate consequence, we get that a cone is closed with respect to taking linear combinations *with nonnegative coefficients* of the elements, and vice versa – a nonempty set closed with respect to taking these combinations is a cone.

Example 1.1.5 *The solution set of an arbitrary (possibly, infinite) system*

$$a_\alpha^T x \leq 0, \ \alpha \in \mathcal{A}$$

of homogeneous linear inequalities with n unknowns x – the set

$$K = \{x : a_\alpha^T x \leq 0 \ \forall \alpha \in \mathcal{A}\}$$

– is a cone.

In particular, the solution set to a homogeneous finite system of m homogeneous linear inequalities

$$Ax \leq 0$$

(A is $m \times n$ matrix) is a cone; a cone of this latter type is called polyhedral.

Note that the cones given by systems of linear homogeneous nonstrict inequalities necessarily are closed. From Separation Theorem 1.2.9 it follows that, vice versa, every closed convex cone is the solution set to such a system, so that Example 1.1.5 is the generic example of a closed convex cone.

Cones form a very important family of convex sets, and one can develop theory of cones absolutely similar (and in a sense, equivalent) to that one of all convex sets. For example, introducing the notion of *conic combination* of vectors $x_1, ..., x_k$ as a linear combination of the vectors with nonnegative coefficients, you can easily prove the following statements completely similar to those for general convex sets, with conic combination playing the role of convex one:

- A set is a cone if and only if it is nonempty and is closed with respect to taking all conic combinations of its elements;
- Intersection of a family of cones is again a cone; in particular, for every nonempty set $M \subset \mathbf{R}^n$ there exists the smallest cone containing M – its conic!hull $\mathrm{Cone}\,(M)$, and this conic hull is comprised of all conic combinations of vectors from M.

In particular, the conic hull of a nonempty finite set $M = \{u_1, ..., u_N\}$ of vectors in $\mathbf{R}^n$ is the cone

$$\mathrm{Cone}\,(M) = \{\sum_{i=1}^N \lambda_i u_i : \lambda_i \geq 0, \ i = 1, ..., N\}.$$

1.1.5 Calculus of convex sets

Proposition 1.1.6 *The following operations preserve convexity of sets:*

1. *Taking intersection: if M_α, $\alpha \in \mathcal{A}$, are convex sets, so is the set $\bigcap\limits_\alpha M_\alpha$.*

2. *Taking direct product: if $M_1 \subset \mathbf{R}^{n_1}$ and $M_2 \subset \mathbf{R}^{n_2}$ are convex sets, so is the set*

$$M_1 \times M_2 = \{y = (y_1, y_2) \in \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} = \mathbf{R}^{n_1+n_2} : y_1 \in M_1, y_2 \in M_2\}.$$

3. *Arithmetic summation and multiplication by reals: if $M_1, ..., M_k$ are convex sets in $\mathbf{R}^n$ and $\lambda_1, ..., \lambda_k$ are arbitrary reals, then the set*

$$\lambda_1 M_1 + ... + \lambda_k M_k = \{\sum_{i=1}^k \lambda_i x_i : x_i \in M_i,\ i = 1, ..., k\}$$

   is convex.

4. *Taking the image under an affine mapping: if $M \subset \mathbf{R}^n$ is convex and $x \mapsto \mathcal{A}(x) \equiv Ax + b$ is an affine mapping from $\mathbf{R}^n$ into $\mathbf{R}^m$ (A is $m \times n$ matrix, b is m-dimensional vector), then the set*

$$\mathcal{A}(M) = \{y = \mathcal{A}(x) \equiv Ax + a : x \in M\}$$

   is a convex set in $\mathbf{R}^m$;

5. *Taking the inverse image under affine mapping: if $M \subset \mathbf{R}^n$ is convex and $y \mapsto Ay + b$ is an affine mapping from $\mathbf{R}^m$ to $\mathbf{R}^n$ (A is $n \times m$ matrix, b is n-dimensional vector), then the set*

$$\mathcal{A}^{-1}(M) = \{y \in \mathbf{R}^m : \mathcal{A}(y) \in M\}$$

   is a convex set in $\mathbf{R}^m$.

Exercise 1.7 *Prove Proposition 1.1.6.*

1.1.6 Topological properties of convex sets

Convex sets and closely related objects - convex functions - play the central role in Optimization. To play this role properly, the convexity alone is insufficient; we need convexity plus closedness.

1.1.6.1 The closure

It is clear from definition of a closed set (Section A.4.3) that the intersection of a family of closed sets in $\mathbf{R}^n$ is also closed. From this fact it, as always, follows that for every subset M of $\mathbf{R}^n$ there exists the smallest closed set containing M; this set is called the *closure* of M and is denoted $\mathrm{cl}\, M$. In Analysis they prove the following inner description of the closure of a set in a metric space (and, in particular, in $\mathbf{R}^n$):

The closure of a set $M \subset \mathbf{R}^n$ is exactly the set comprised of the limits of all converging sequences of elements of M.

With this fact in mind, it is easy to prove that, e.g., the closure of the open Euclidean ball

$$\{x : |x - a| < r\} \quad [r > 0]$$

is the closed ball $\{x :\| x - a \|_2 \leq r\}$. Another useful application example is the closure of a set

$$M = \{x : a_\alpha^T x < b_\alpha,\ \alpha \in \mathcal{A}\}$$

given by strict linear inequalities: *if such a set is nonempty*, then its closure is given by the nonstrict versions of the same inequalities:

$$\mathrm{cl}\, M = \{x : a_\alpha^T x \leq b_\alpha,\ \alpha \in \mathcal{A}\}.$$

Nonemptiness of M in the latter example is essential: the set M given by two strict inequalities

$$x < 0, \quad -x < 0$$

in $\mathbf{R}$ clearly is empty, so that its closure also is empty; in contrast to this, applying formally the above rule, we would get wrong answer

$$\mathrm{cl}\, M = \{x : x \leq 0, x \geq 0\} = \{0\}.$$

1.1.6.2 The interior

Let $M \subset \mathbf{R}^n$. We say that a point $x \in M$ is an *interior* point of M, if some neighbourhood of the point is contained in M, i.e., there exists centered at x ball of positive radius which belongs to M:

$$\exists r > 0 \quad B_r(x) \equiv \{y :\| y - x \|_2 \leq r\} \subset M.$$

The set of all interior points of M is called the *interior* of M [notation: int M].

For example,

- The interior of an open set is the set itself;
- The interior of the closed ball $\{x :\| x - a \|_2 \leq r\}$ is the open ball $\{x :\| x - a \|_2 < r\}$ (why?)
- The interior of a polyhedral set $\{x : Ax \leq b\}$ with matrix A not containing zero rows is the set $\{x : Ax < b\}$ (why?)

  The latter statement is not, generally speaking, valid for sets of solutions of infinite systems of linear inequalities. For example, the system of inequalities

  $$x \leq \frac{1}{n},\ n = 1, 2, ...$$

  in $\mathbf{R}$ has, as a solution set, the nonpositive ray $\mathbf{R}_- = \{x \leq 0\}$; the interior of this ray is the negative ray $\{x < 0\}$. At the same time, strict versions of our inequalities

  $$x < \frac{1}{n},\ n = 1, 2, ...$$

  define the same nonpositive ray, not the negative one.

It is also easily seen (this fact is valid for arbitrary metric spaces, not for $\mathbf{R}^n$ only), that

- the interior of an arbitrary set is open

The interior of a set is, of course, contained in the set, which, in turn, is contained in its closure:

$$\text{int } M \subset M \subset \text{cl } M. \tag{1.1.1}$$

The complement of the interior in the closure – the set

$$\partial M = \text{cl } M \setminus \text{int } M$$

– is called the *boundary* of M, and the points of the boundary are called *boundary points* of M (Warning: these points not necessarily belong to M, since M can be less than $\text{cl } M$; in fact, all boundary points belong to M if and only if $M = \text{cl } M$, i.e., if and only if M is closed).

The boundary of a set clearly is closed (as the intersection of two closed sets $\text{cl } M$ and $\mathbf{R}^n \setminus \text{int } M$; the latter set is closed as a complement to an open set). From the definition of the boundary,

$$M \subset \text{int } M \cup \partial M \quad [= \text{cl } M],$$

so that a point from M is either an interior, or a boundary point of M.

1.1.6.3 The relative interior

Many of the constructions in Optimization possess nice properties in the interior of the set the construction is related to and may lose these nice properties at the boundary points of the set; this is why in many cases we are especially interested in interior points of sets and want the set of these points to be "enough massive". What to do if it is not the case – e.g., there are no interior points at all (look at a segment in the plane)? It turns out that in these cases we can use a good surrogate of the "normal" interior – the *relative interior* defined as follows.

Definition 1.1.3 [Relative interior] *Let $M \subset \mathbf{R}^n$. We say that a point $x \in M$ is relative interior for M, if M contains the intersection of a small enough ball centered at x with $\text{Aff}(M)$:*

$$\exists r > 0 \quad B_r(x) \cap \text{Aff}(M) \equiv \{y : y \in \text{Aff}(M),\ \| y - x \|_2 \leq r\} \subset M.$$

The set of all relative interior points of M is called its relative interior [notation: $\text{rint } M$].

For example the relative interior of a singleton is the singleton itself (since a point in the 0-dimensional space is the same as a ball of a positive radius); more generally, the relative interior of an affine subspace is the set itself. The interior of a segment $[x, y]$ $(x \neq y)$ in $\mathbf{R}^n$ is empty whenever $n > 1$; in contrast to this, the relative interior is nonempty independently of n and is the interval (x, y) – the segment with deleted endpoints. Geometrically speaking, the relative interior is the interior we get when regard M as a subset of its affine hull (the latter, geometrically, is nothing but $\mathbf{R}^k$, k being the affine dimension of $\text{Aff}(M)$).

Exercise 1.8 *Prove that the relative interior of a simplex with vertices $y_0, ..., y_m$ is exactly the set $\{x = \sum_{i=0}^{m} \lambda_i y_i : \lambda_i > 0, \sum_{i=0}^{m} \lambda_i = 1\}$.*

We can play with the notion of the relative interior in basically the same way as with the one of interior, namely:

- since $\text{Aff}(M)$, as every affine subspace, is closed and contains M, it contains also the smallest closed sets containing M, i.e., $\text{cl } M$. Therefore we have the following analogies of inclusions (1.1.1):

$$\text{rint } M \subset M \subset \text{cl } M \quad [\subset \text{Aff}(M)]; \tag{1.1.2}$$

- we can define the *relative boundary* $\partial_{\rm ri} M = {\rm cl}\, M \backslash {\rm rint}\, M$ which is a closed set contained in ${\rm Aff}(M)$, and, as for the "actual" interior and boundary, we have

$$ {\rm rint}\, M \subset M \subset {\rm cl}\, M = {\rm rint}\, M + \partial_{\rm ri} M. $$

Of course, if ${\rm Aff}(M) = \mathbf{R}^n$, then the relative interior becomes the usual interior, and similarly for boundary; this for sure is the case when ${\rm int}\, M \neq \emptyset$ (since then M contains a ball B, and therefore the affine hull of M is the entire $\mathbf{R}^n$, which is the affine hull of B).

1.1.6.4 Nice topological properties of convex sets

An arbitrary set M in $\mathbf{R}^n$ may possess very pathological topology: both inclusions in the chain

$$ {\rm rint}\, M \subset M \subset {\rm cl}\, M $$

can be very "non-tight". For example, let M be the set of rational numbers in the segment $[0,1] \subset \mathbf{R}$. Then ${\rm rint}\, M = {\rm int}\, M = \emptyset$ – since every neighbourhood of every rational real contains irrational reals – while ${\rm cl}\, M = [0,1]$. Thus, ${\rm rint}\, M$ is "incomparably smaller" than M, ${\rm cl}\, M$ is "incomparably larger", and M is contained in its relative boundary (by the way, what is this relative boundary?).

The following proposition demonstrates that the topology of a *convex* set M is much better than it might be for an arbitrary set.

Theorem 1.1.1 *Let M be a convex set in $\mathbf{R}^n$. Then*

(i) *The interior* ${\rm int}\, M$*, the closure* ${\rm cl}\, M$ *and the relative interior* ${\rm rint}\, M$ *are convex;*

(ii) *If M is nonempty, then the relative interior* ${\rm rint}\, M$ *of M is nonempty*

(iii) *The closure of M is the same as the closure of its relative interior:*

$$ {\rm cl}\, M = {\rm cl}\, {\rm rint}\, M $$

(in particular, every point of ${\rm cl}\, M$ *is the limit of a sequence of points from* ${\rm rint}\, M$*)*

(iv) *The relative interior remains unchanged when we replace M with its closure:*

$$ {\rm rint}\, M = {\rm rint}\ {\rm cl}\, M. $$

Proof. (i): prove yourself!

(ii): Let M be a nonempty convex set, and let us prove that ${\rm rint}\, M \neq \emptyset$. By translation, we may assume that $0 \in M$. Further, we may assume that the linear span of M is the entire $\mathbf{R}^n$. Indeed, as far as linear operations and the Euclidean structure are concerned, the linear span L of M, as every other linear subspace in $\mathbf{R}^n$, is equivalent to certain $\mathbf{R}^k$; since the notion of relative interior deals only with linear and Euclidean structures, we lose nothing thinking of ${\rm Lin}(M)$ as of $\mathbf{R}^k$ and taking it as our universe instead of the original universe $\mathbf{R}^n$. Thus, in the rest of the proof of (ii) we assume that $0 \in M$ and ${\rm Lin}(M) = \mathbf{R}^n$; what we should prove is that the interior of M (which in the case in question is the same as relative interior) is nonempty. Note that since $0 \in M$, we have ${\rm Aff}(M) = {\rm Lin}(M) = \mathbf{R}^n$.

Since ${\rm Lin}(M) = \mathbf{R}^n$, we can find in M n linearly independent vectors $a_1, .., a_n$. Let also $a_0 = 0$. The $n+1$ vectors $a_0, ..., a_n$ belong to M, and *since M is convex*, the convex hull of these vectors also belongs to M. This convex hull is the set

$$ \Delta = \{x = \sum_{i=0}^{n} \lambda_i a_i : \lambda \geq 0, \sum_i \lambda_i = 1\} = \{x = \sum_{i=1}^{n} \mu_i a_i : \mu \geq 0, \sum_{i=1}^{n} \mu_i \leq 1\}. $$

We see that Δ is the image of the *standard full-dimensional simplex*

$$\{\mu \in \mathbf{R}^n : \mu \geq 0, \sum_{i=1}^{n} \mu_i \leq 1\}$$

under linear transformation $\mu \mapsto A\mu$, where A is the matrix with the columns $a_1, ..., a_n$. The standard simplex clearly has a nonempty interior (comprised of all vectors $\mu > 0$ with $\sum_i \mu_i < 1$); since A is nonsingular (due to linear independence of $a_1, ..., a_n$), multiplication by A maps open sets onto open ones, so that Δ has a nonempty interior. Since $\Delta \subset M$, the interior of M is nonempty. □

(iii): We should prove that the closure of $\operatorname{rint} M$ is exactly the same that the closure of M. In fact we shall prove even more:

Lemma 1.1.1 *Let $x \in \operatorname{rint} M$ and $y \in \operatorname{cl} M$. Then all points from the half-segment $[x, y)$,*

$$[x, y) = \{z = (1 - \lambda)x + \lambda y : 0 \leq \lambda < 1\}$$

belong to the relative interior of M.

Proof of the Lemma. Let $\operatorname{Aff}(M) = a + L$, L being linear subspace; then

$$M \subset \operatorname{Aff}(M) = x + L.$$

Let B be the unit ball in L:

$$B = \{h \in L : \| h \|_2 \leq 1\}.$$

Since $x \in \operatorname{rint} M$, there exists positive radius r such that

$$x + rB \subset M. \tag{1.1.3}$$

Now let $\lambda \in [0, 1)$, and let $z = (1 - \lambda)x + \lambda y$. Since $y \in \operatorname{cl} M$, we have $y = \lim_{i \to \infty} y_i$ for certain sequence of points from M. Setting $z_i = (1 - \lambda)x + \lambda y_i$, we get $z_i \to z$ as $i \to \infty$. Now, from (1.1.3) and the convexity of M is follows that the sets $Z_i = \{u = (1 - \lambda)x' + \lambda y_i : x' \in x + rB\}$ are contained in M; clearly, Z_i is exactly the set $z_i + r'B$, where $r' = (1 - \lambda)r > 0$. Thus, z is the limit of sequence z_i, and r'-neighbourhood (in $\operatorname{Aff}(M)$) of every one of the points z_i belongs to M. For every $r'' < r'$ and for all i such that z_i is close enough to z, the r'-neighbourhood of z_i contains the r''-neighbourhood of z; thus, a neighbourhood (in $\operatorname{Aff}(M)$) of z belongs to M, whence $z \in \operatorname{rint} M$. ■

A useful byproduct of Lemma 1.1.1 is as follows:

Corollary 1.1.2 *Let M be a convex set. Then every convex combination*

$$\sum_i \lambda_i x_i$$

of points $x_i \in \operatorname{cl} M$ where at least one term with positive coefficient corresponds to $x_i \in \operatorname{rint} M$ is in fact a point from $\operatorname{rint} M$.

(iv): The statement is evidently true when M is empty, so assume that M is nonempty. The inclusion $\operatorname{rint} M \subset \operatorname{rint} \operatorname{cl} M$ is evident, and all we need is to prove the inverse inclusion. Thus, let $z \in \operatorname{rint} \operatorname{cl} M$, and let us prove that $z \in \operatorname{rint} M$. Let $x \in \operatorname{rint} M$ (we already know that the latter set is nonempty). Consider the segment $[x, z]$; since z is in the relative interior of $\operatorname{cl} M$, we can extend a little bit this segment through the point z, not leaving $\operatorname{cl} M$, i.e., there exists $y \in \operatorname{cl} M$ such that $z \in [x, y)$. We are done, since by Lemma 1.1.1 from $z \in [x, y)$, with $x \in \operatorname{rint} M$, $y \in \operatorname{cl} M$, it follows that $z \in \operatorname{rint} M$. ■

We see from the proof of Theorem 1.1.1 that to get a closure of a (nonempty) convex set, it suffices to subject it to the "radial" closure, i.e., to take a point $x \in \operatorname{rint} M$, take all rays in $\operatorname{Aff}(M)$ starting at x and look at the intersection of such a ray l with M; such an intersection will be a convex set on the line which contains a one-sided neighbourhood of x, i.e., is either a segment $[x, y_l]$, or the entire ray l, or a half-interval $[x, y_l)$. In the first two cases we should not do anything; in the third we should add y to M. After all rays are looked through and all "missed" endpoints y_l are added to M, we get the closure of M. To understand what is the role of convexity here, look at the *nonconvex* set of rational numbers from $[0, 1]$; the interior ($\equiv$ relative interior) of this "highly percolated" set is empty, the closure is $[0, 1]$, and there is no way to restore the closure in terms of the interior.

1.2 Main theorems on convex sets

1.2.1 Caratheodory Theorem

Let us call the *affine dimension* (or simple *dimension* of a nonempty set $M \subset \mathbf{R}^n$ (notation: $\dim M$) the affine dimension of $\operatorname{Aff}(M)$.

Theorem 1.2.1 [Caratheodory] *Let $M \subset \mathbf{R}^n$, and let $\dim \operatorname{Conv} M = m$. Then every point $x \in \operatorname{Conv} M$ is a convex combination of at most $m + 1$ points from M.*

Proof. Let $x \in \operatorname{Conv} M$. By Proposition 1.1.4 on the structure of convex hull, x is convex combination of certain points $x_1, ..., x_N$ from M:

$$x = \sum_{i=1}^{N} \lambda_i x_i, \quad [\lambda_i \geq 0, \sum_{i=1}^{N} \lambda_i = 1].$$

Let us choose among all these representations of x as a convex combination of points from M the one with the smallest possible N, and let it be the above combination. We claim that $N \leq m + 1$ (this claim leads to the desired statement). Indeed, if $N > m + 1$, then the system of $m + 1$ homogeneous equations

$$\begin{aligned} \sum_{i=1}^{N} \mu_i x_i &= 0 \\ \sum_{i=1}^{N} \mu_i &= 0 \end{aligned}$$

with N unknowns $\mu_1, ..., \mu_N$ has a nontrivial solution $\delta_1, ..., \delta_N$:

$$\sum_{i=1}^{N} \delta_i x_i = 0, \sum_{i=1}^{N} \delta_i = 0, (\delta_1, ..., \delta_N) \neq 0.$$

It follows that, for every real t,

$$(*) \quad \sum_{i=1}^{N} [\lambda_i + t\delta_i] x_i = x.$$

What is to the left, is an affine combination of x_i's. When $t = 0$, this is a convex combination - all coefficients are nonnegative. When t is large, this is *not* a convex combination, since some of δ_i's are negative (indeed, not all of them are zero, and the sum of δ_i's is 0). There exists, of course, the largest t for which the combination (*) has nonnegative coefficients, namely

$$t^* = \min_{i:\delta_i < 0} \frac{\lambda_i}{|\delta_i|}.$$

For this value of t, the combination (*) is with nonnegative coefficients, and at least one of the coefficients is zero; thus, we have represented x as a convex combination of less than N points from M, which contradicts the definition of N. ∎

1.2.2 Radon Theorem

Theorem 1.2.2 [Radon] *Let S be a set of at least $n+2$ points $x_1, ..., x_N$ in $\mathbf{R}^n$. Then one can split the set into two nonempty subsets S_1 and S_2 with intersecting convex hulls: there exists partitioning $I \cup J = \{1, ..., N\}$, $I \cap J = \emptyset$, of the index set $\{1, ..., N\}$ into two nonempty sets I and J and convex combinations of the points $\{x_i, i \in I\}$, $\{x_j, j \in J\}$ which coincide with each other, i.e., there exist α_i, $i \in I$, and β_j, $j \in J$, such that*

$$\sum_{i\in I} \alpha_i x_i = \sum_{j\in J} \beta_j x_j; \quad \sum_i \alpha_i = \sum_j \beta_j = 1; \quad \alpha_i, \beta_j \geq 0.$$

Proof. Since $N > n+1$, the homogeneous system of $n+1$ scalar equations with N unknowns $\mu_1, ..., \mu_N$

$$\begin{aligned} \sum_{i=1}^N \mu_i x_i &= 0 \\ \sum_{i=1}^N \mu_i &= 0 \end{aligned}$$

has a nontrivial solution $\lambda_1, ..., \lambda_N$:

$$\sum_{i=1}^N \mu_i x_i = 0, \sum_{i=1}^N \lambda_i = 0, [(\lambda_1, ..., \lambda_N) \neq 0].$$

Let $I = \{i : \lambda_i \geq 0\}$, $J = \{i : \lambda_i < 0\}$; then I and J are nonempty and form a partitioning of $\{1, ..., N\}$. We have

$$a \equiv \sum_{i\in I} \lambda_i = \sum_{j\in J} (-\lambda_j) > 0$$

(since the sum of all λ's is zero and not all λ's are zero). Setting

$$\alpha_i = \frac{\lambda_i}{a}, i \in I, \ \beta_j = \frac{-\lambda_j}{a}, j \in J,$$

we get

$$\alpha_i \geq 0, \beta_j \geq 0, \sum_{i\in I} \alpha_i = 1, \sum_{j\in J} \beta_j = 1,$$

and

$$[\sum_{i\in I} \alpha_i x_i] - [\sum_{j\in J} \beta_j x_j] = a^{-1} \left([\sum_{i\in I} \lambda_i x_i] - [\sum_{j\in J} (-\lambda_j) x_j] \right) = a^{-1} \sum_{i=1}^N \lambda_i x_i = 0. \quad \blacksquare$$

1.2.3 Helley Theorem

Theorem 1.2.3 [Helley, I] *Let $\mathcal{F}$ be a finite family of convex sets in $\mathbf{R}^n$. Assume that every $n+1$ sets from the family have a point in common. Then all the sets have a point in common.*

Proof. Let us prove the statement by induction on the number N of sets in the family. The case of $N \leq n+1$ is evident. Now assume that the statement holds true for all families with certain number $N \geq n+1$ of sets, and let $S_1, ..., S_N, S_{N+1}$ be a family of $N+1$ convex sets which satisfies the premise of the Helley Theorem; we should prove that the intersection of the sets $S_1, ..., S_N, S_{N+1}$ is nonempty.

Deleting from our $N+1$-set family the set S_i, we get N-set family which satisfies the premise of the Helley Theorem and thus, by the inductive hypothesis, the intersection of its members is nonempty:

$$(\forall i \leq N+1): \ T^i = S_1 \cap S_2 \cap ... \cap S_{i-1} \cap S_{i+1} \cap ... \cap S_{N+1} \neq \emptyset.$$

Let us choose a point x_i in the (nonempty) set T^i. We get $N+1 \geq n+2$ points from $\mathbf{R}^n$. By Radon's Theorem, we can partition the index set $\{1,...,N+1\}$ into two nonempty subsets I and J in such a way that certain convex combination x of the points x_i, $i \in I$, is a convex combination of the points x_j, $j \in J$, as well. Let us verify that x belongs to all the sets $S_1,...,S_{N+1}$, which will complete the proof. Indeed, let i^* be an index from our index set; let us prove that $x \in S_{i^*}$. We have either $i^* \in I$, or $i^* \in J$. In the first case all the sets T^j, $j \in J$, are contained in S_{i^*} (since S_{i^*} participates in all intersections which give T^i with $i \neq i^*$). Consequently, all the points x_j, $j \in J$, belong to S_{i^*}, and therefore x, which is a convex combination of these points, also belongs to S_{i^*} (all our sets are convex!), as required. In the second case similar reasoning says that all the points x_i, $i \in I$, belong to S_{i^*}, and therefore x, which is a convex combination of these points, belongs to S_{i^*}. ∎

Exercise 1.9 *Let $S_1,...,S_N$ be a family of N convex sets in $\mathbf{R}^n$, and let m be the affine dimension of $\mathrm{Aff}(S_1 \cup ... \cup S_N)$. Assume that every $m+1$ sets from the family have a point in common. Prove that all sets from the family have a point in common.*

In the aforementioned version of the Helley Theorem we dealt with finite families of convex sets. To extend the statement to the case of infinite families, we need to strengthen slightly the assumption. The resulting statement is as follows:

Theorem 1.2.4 [Helley, II] *Let $\mathcal{F}$ be an arbitrary family of convex sets in $\mathbf{R}^n$. Assume that*

(a) every $n+1$ sets from the family have a point in common,

and

(b) every set in the family is closed, and the intersection of the sets from certain finite subfamily of the family is bounded (e.g., one of the sets in the family is bounded).

Then all the sets from the family have a point in common.

Proof. By the previous theorem, all finite subfamilies of $\mathcal{F}$ have nonempty intersections, and these intersections are convex (since intersection of a family of convex sets is convex, Theorem 1.1.3); in view of (a) these intersections are also closed. Adding to $\mathcal{F}$ all intersections of finite subfamilies of $\mathcal{F}$, we get a larger family $\mathcal{F}'$ comprised of closed convex sets, and a finite subfamily of this larger family again has a nonempty intersection. Besides this, from (b) it follows that this new family contains a bounded set Q. Since all the sets are closed, the family of sets

$$\{Q \cap Q' : Q' \in \mathcal{F}\}$$

is a *nested family of compact sets* (i.e., a family of compact sets with nonempty intersection of sets from every finite subfamily); by the well-known Analysis theorem such a family has a nonempty intersection[1)].

∎

1.2.4 Polyhedral representations and Fourier-Motzkin Elimination

1.2.4.1 Polyhedral representations

Recall that by definition a polyhedral set X in $\mathbf{R}^n$ is the set of solutions of a finite system of nonstrict linear inequalities in variables $x \in \mathbf{R}^n$:

$$X = \{x \in \mathbf{R}^n : Ax \leq b\} = \{x \in \mathbf{R}^n : a_i^T x \leq b_i,\ 1 \leq i \leq m\}.$$

We shall call such a representation of X its *polyhedral description*. A polyhedral set always is convex and closed (Proposition 1.1.1). Now let us introduce the notion of *polyhedral representation* of a set $X \subset \mathbf{R}^n$.

[1)] here is the proof of this Analysis theorem: assume, on contrary, that the compact sets Q_α, $\alpha \in \mathcal{A}$, have empty intersection. Choose a set Q_{α^*} from the family; for every $x \in Q_{\alpha^*}$ there is a set Q^x in the family which does not contain x - otherwise x would be a common point of all our sets. Since Q^x is closed, there is an open ball V_x centered at x which does not intersect Q^x. The balls V_x, $x \in Q_{\alpha^*}$, form an open covering of the compact set Q_{α^*}, and therefore there exists a finite subcovering $V_{x_1},...,V_{x_N}$ of Q_{α^*} by the balls from the covering. Since Q^{x_i} does not intersect V_{x_i}, we conclude that the intersection of the finite subfamily $Q_{\alpha^*}, Q^{x_1},...,Q^{x_N}$ is empty, which is a contradiction

Definition 1.2.1 We say that a set $X \subset \mathbf{R}^n$ is *polyhedrally representable*, if it admits a representation as follows:

$$X = \{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^k : Ax + Bu \leq c\} \tag{1.2.1}$$

where A, B are $m \times n$ and $m \times k$ matrices and $c \in \mathbf{R}^m$. A representation of X of the form of (1.2.1) is called a *polyhedral representation of* X, and variables u in such a representation are called *slack variables*.

Geometrically, a polyhedral representation of a set $X \subset \mathbf{R}^n$ is its representation as the *projection* $\{x : \exists u : (x,u) \in Y\}$ of a *polyhedral* set $Y = \{(x,u) : Ax + Bu \leq c\}$ in the space of $n+k$ variables $(x \in \mathbf{R}^n, u \in \mathbf{R}^k)$ under the linear mapping (the projection) $(x,u) \mapsto x : \mathbf{R}^{n+k}_{x,u} \to \mathbf{R}^n_x$ of the $n+k$-dimensional space of (x,u)-variables (the space where Y lives) to the n-dimensional space of x-variables where X lives.

Note that every polyhedrally representable set is the image under linear mapping (even a projection) of a polyhedral, and thus convex, set. It follows that *a polyhedrally representable set definitely is convex* (Proposition 1.1.6).

Examples: 1) Every polyhedral set $X = \{x \in \mathbf{R}^n : Ax \leq b\}$ is polyhedrally representable – a polyhedral description of X is nothing but a polyhedral representation with no slack variables ($k = 0$). Vice versa, a polyhedral representation of a set X with no slack variables ($k = 0$) clearly is a polyhedral description of the set (which therefore is polyhedral).

2) Looking at the set $X = \{x \in \mathbf{R}^n : \sum_{i=1}^n |x_i| \leq 1\}$, we cannot say immediately whether it is or is not polyhedral; at least the initial description of X is *not* of the form $\{x : Ax \leq b\}$. However, X admits a polyhedral representation, e.g., the representation

$$X = \{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^n : \underbrace{-u_i \leq x_i \leq u_i}_{\Leftrightarrow |x_i| \leq u_i}, 1 \leq i \leq n, \sum_{i=1}^n u_i \leq 1\}. \tag{1.2.2}$$

Note that the set X in question can be described by a system of linear inequalities *in x-variables only*, namely, as

$$X = \{x \in \mathbf{R}^n : \sum_{i=1}^n \epsilon_i x_i \leq 1 \, , \forall (\epsilon_i = \pm 1, \, 1 \leq i \leq n)\},$$

that is, X is polyhedral. However, the above polyhedral description of X (which in fact is minimal in terms of the number of inequalities involved) requires 2^n inequalities — an astronomically large number when n is just few tens. In contrast to this, the polyhedral representation (1.2.2) of the same set requires just n slack variables u and $2n+1$ linear inequalities on x, u – the "complexity" of this representation is just linear in n.

3) Let $a_1, ..., a_m$ be given vectors in $\mathbf{R}^n$. Consider the conic hull of the finite set $\{a_1, ..., a_m\}$ – the set $\text{Cone}\,\{a_1, ..., a_m\} = \{x = \sum_{i=1}^m \lambda_i a_i : \lambda \geq 0\}$ (see section 1.1.4). It is absolutely unclear whether this set is polyhedral. In contrast to this, its polyhedral representation is immediate:

$$\begin{aligned} \text{Cone}\,\{a_1, ..., a_m\} = \; & \{x \in \mathbf{R}^n : \exists \lambda \geq 0 : x = \textstyle\sum_{i=1}^m \lambda_i a_i\} \\ = \; & \{x \in \mathbf{R}^n : \exists \lambda \in \mathbf{R}^m : \left\{ \begin{array}{l} -\lambda \leq 0 \\ x - \sum_{i=1}^m \lambda_i a_i \leq 0 \\ -x + \sum_{i=1}^m \lambda_i a_i \leq 0 \end{array} \right. \} \end{aligned}$$

In other words, the original description of X is nothing but its polyhedral representation (in slight disguise), with λ_i's in the role of slack variables.

1.2.4.2 Every polyhedrally representable set is polyhedral! (Fourier-Motzkin elimination)

A surprising and deep fact is that the situation in the Example 2) above is quite general:

Theorem 1.2.5 *Every polyhedrally representable set is polyhedral.*

Proof: Fourier-Motzkin Elimination. Recalling the definition of a polyhedrally representable set, our claim can be rephrased equivalently as follows: *the projection of a polyhedral set Y in a space $\mathbf{R}^{n+k}_{x,u}$ of (x,u)-variables on the subspace $\mathbf{R}^n_x$ of x-variables is a polyhedral set in $\mathbf{R}^n$*. All we need is to prove this claim in the case of exactly one slack variable (since the projection which reduces the dimension by k — "kills k slack variables" — is the result of k subsequent projections, every one reducing the dimension by 1 ("killing one slack variable each")).

Thus, let

$$Y = \{(x,u) \in \mathbf{R}^{n+1} : a_i^T x + b_i u \le c_i, 1 \le i \le m\}$$

be a polyhedral set with n variables x and single variable u; we want to prove that the projection

$$X = \{x : \exists u : Ax + bu \le c\}$$

of Y on the space of x-variables is polyhedral. To see it, let us split the inequalities defining Y into three groups (some of them can be empty):
— "black" inequalities — those with $b_i = 0$; these inequalities do not involve u at all;
— "red" inequalities – those with $b_i > 0$. Such an inequality can be rewritten equivalently as $u \le b_i^{-1}[c_i - a_i^T x]$, and it imposes a (depending on x) *upper bound* on u;
— "green" inequalities – those with $b_i < 0$. Such an inequality can be rewritten equivalently as $u \ge b_i^{-1}[c_i - a_i^T x]$, and it imposes a (depending on x) *lower bound* on u.
Now it is clear when $x \in X$, that is, when x can be extended, by some u, to a point (x,u) from Y: this is the case if and only if, first, x satisfies all black inequalities, and, second, the red upper bounds on u specified by x are compatible with the green lower bounds on u specified by x, meaning that every lower bound is $\le$ every upper bound (the latter is necessary and sufficient to be able to find a value of u which is $\ge$ all lower bounds and $\le$ all upper bounds). Thus,

$$X = \left\{ x : \left\{ \begin{array}{l} a_i^T x \le c_i \text{ for all "black" indexes } i \text{ – those with } b_i = 0 \\ b_j^{-1}[c_j - a_j^T x] \le b_k^{-1}[c_k - a_k^T x] \text{ for all "green" (i.e., with } b_j < 0) \text{ indexes } j \\ \quad \text{and all "red" (i.e., with } b_k > 0) \text{ indexes } k \end{array} \right\} \right.$$

We see that X is given by finitely many nonstrict linear inequalities in x-variables only, as claimed. ∎

The outlined procedure for building polyhedral descriptions (i.e., polyhedral representations not involving slack variables) for projections of polyhedral sets is called *Fourier-Motzkin elimination*.

1.2.4.3 Some applications

As an immediate application of Fourier-Motzkin elimination, let us take a linear program $\min_x \{c^T x : Ax \le b\}$ and look at the set T of values of the objective at all feasible solutions, if any:

$$T = \{t \in \mathbf{R} : \exists x : c^T x = t, Ax \le b\}.$$

Rewriting the linear equality $c^T x = t$ as a pair of opposite inequalities, we see that T is polyhedrally representable, and the above definition of T is nothing but a polyhedral representation of this set, with x in the role of the vector of slack variables. By Fourier-Motzkin elimination, T is polyhedral – this set is given by a finite system of nonstrict linear inequalities in variable t only. As such, as it is immediately seen, T is
— either empty (meaning that the LP in question is infeasible),
— or is a below unbounded nonempty set of the form $\{t \in \mathbf{R} : -\infty \le t \le b\}$ with $b \in \mathbf{R} \cup \{+\infty\}$ (meaning that the LP is feasible and unbounded),
— or is a below bounded nonempty set of the form $\{t \in \mathbf{R} : -a \le t \le b\}$ with $a \in \mathbf{R}$ and $+\infty \ge b \ge a$. In this case, the LP is feasible and bounded, and a is its optimal value.
Note that given the list of linear inequalities defining T (this list can be built algorithmically by Fourier-Motzkin elimination as applied to the original polyhedral representation of T), we can easily detect which one of the above cases indeed takes place, i.e., to identify the feasibility and boundedness status of the LP and to find its optimal value. When it is finite (case 3 above), we can use the Fourier-Motzkin

elimination backward, starting with $t = a \in T$ and extending this value to a pair (t, x) with $t = a = c^T x$ and $Ax \leq b$, that is, we can augment the optimal value by an optimal solution. Thus, we can say that Fourier-Motzkin elimination is a finite Real Arithmetics algorithm which allows to check whether an LP is feasible and bounded, and when it is the case, allows to find the optimal value and an optimal solution. An unpleasant fact of life is that this algorithm is completely impractical, since the elimination process can blow up exponentially the number of inequalities. Indeed, from the description of the process it is clear that if a polyhedral set is given by m linear inequalities, then eliminating one variable, we can end up with as much as $m^2/4$ inequalities (this is what happens if there are $m/2$ red, $m/2$ green and no black inequalities). Eliminating the next variable, we again can "nearly square" the number of inequalities, and so on. Thus, the number of inequalities in the description of T can become astronomically large when even when the dimension of x is something like 10. The actual importance of Fourier-Motzkin elimination is of theoretical nature. For example, the LP-related reasoning we have just carried out shows that *every feasible and bounded LP program is solvable – has an optimal solution* (we shall revisit this result in more details in section 1.2.9.2. This is a fundamental fact for LP, and the above reasoning (even with the justification of the elimination "charged" to it) is the shortest and most transparent way to prove this fundamental fact. Another application of the fact that polyhedrally representable sets are polyhedral is the Homogeneous Farkas Lemma to be stated and proved in section 1.2.5.A; this lemma will be instrumental in numerous subsequent theoretical developments.

1.2.4.4 Calculus of polyhedral representations

The fact that polyhedral sets are exactly the same as polyhedrally representable ones does not nullify the notion of a polyhedral representation. The point is that a set can admit "quite compact" polyhedral representation involving slack variables and require astronomically large, completely meaningless for any practical purpose number of inequalities in its polyhedral description (think about the set (1.2.2) when $n = 100$). Moreover, polyhedral representations admit a kind of "fully algorithmic calculus." Specifically, it turns out that all basic convexity-preserving operations (cf. Proposition 1.1.6) as applied to polyhedral operands preserve polyhedrality; moreover, polyhedral representations of the results are readily given by polyhedral representations of the operands. Here is "algorithmic polyhedral analogy" of Proposition 1.1.6:

1. *Taking finite intersection*: Let M_i, $1 \leq i \leq m$, be polyhedral sets in $\mathbf{R}^n$ given by their polyhedral representations
$$M_i = \{x \in \mathbf{R}^n : \exists u^i \in \mathbf{R}^{k_i} : A_i x + B_i u^i \leq c^i\},\ 1 \leq i \leq m.$$
Then the intersection of the sets M_i is polyhedral with an explicit polyhedral representation, specifically,
$$\bigcap_{i=1}^m M_i = \{x \in \mathbf{R}^n : \exists u = (u^1, ..., u^m) \in \mathbf{R}^{k_1+...+k_m} : \underbrace{A_i x + B_i u^i \leq c^i,\ 1 \leq i \leq m}_{\text{system of nonstrict linear inequalities in } x, u}\}$$

2. *Taking direct product*: Let $M_i \subset \mathbf{R}^{n_i}$, $1 \leq i \leq m$, be polyhedral sets given by polyhedral representations
$$M_i = \{x^i \in \mathbf{R}^{n_i} : \exists u^i \in \mathbf{R}^{k_i} : A_i x^i + B_i u^i \leq c^i\},\ 1 \leq i \leq m.$$
Then the direct product $M_1 \times ... \times M_m := \{x = (x^1, ..., x^m) : x^i \in M_i,\ 1 \leq i \leq m\}$ of the sets is a polyhedral set with explicit polyhedral representation, specifically,
$$\begin{array}{rl} M_1 \times ... \times M_m = & \{x = (x^1, ..., x^m) \in \mathbf{R}^{n_1+...+n_m} : \\ & \exists u = (u^1, ..., u^m) \in \mathbf{R}^{k_1+...+k_m} : A_i x^i + B_i u^i \leq c^i,\ 1 \leq i \leq m\}. \end{array}$$

3. *Arithmetic summation and multiplication by reals*: Let $M_i \subset \mathbf{R}^n$, $1 \leq i \leq m$, be polyhedral sets given by polyhedral representations
$$M_i = \{x \in \mathbf{R}^n : \exists u^i \in \mathbf{R}^{k_i} : A_i x + B_i u^i \leq c^i\},\ 1 \leq i \leq m,$$

and let $\lambda_1, ..., \lambda_k$ be reals. Then the set $\lambda_1 M_1 + ... + \lambda_m M_m := \{x = \lambda_1 x_1 + ... + \lambda_m x_m : x_i \in M_i,\ 1 \le i \le m\}$ is polyhedral with explicit polyhedral representation, specifically,

$$\begin{array}{rl}\lambda_1 M_1 + ... + \lambda_m M_m = & \{x \in \mathbf{R}^n : \exists (x^i \in \mathbf{R}^n, u^i \in \mathbf{R}^{k_i}, 1 \le i \le m) : \\ & x \le \sum_i \lambda_i x^i, x \ge \sum_i \lambda_i x^i,\ A_i x^i + B_i u^i \le c^i,\ 1 \le i \le m\}.\end{array}$$

4. Taking the image under an affine mapping: Let $M \subset \mathbf{R}^n$ be a polyhedral set given by polyhedral representation
$$M = \{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^k : Ax + Bu \le c\}$$
and let $\mathcal{P}(x) = Px + p : \mathbf{R}^n \to \mathbf{R}^m$ be an affine mapping. Then the image $\mathcal{P}(M) := \{y = Px + p : x \in M\}$ of M under the mapping is polyhedral set with explicit polyhedral representation, specifically,
$$\mathcal{P}(M) = \{y \in \mathbf{R}^m : \exists (x \in \mathbf{R}^n, u \in \mathbf{R}^k) : y \le Px + p, y \ge Px + p, Ax + Bu \le c\}.$$

5. Taking the inverse image under affine mapping: Let $M \subset \mathbf{R}^n$ be polyhedral set given by polyhedral representation
$$M = \{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^k : Ax + Bu \le c\}$$
and let $\mathcal{P}(y) = Py + p : \mathbf{R}^m \to \mathbf{R}^n$ be an affine mapping. Then the inverse image $\mathcal{P}^{-1}(M) := \{y : Py + p \in M\}$ of M under the mapping is polyhedral set with explicit polyhedral representation, specifically,
$$\mathcal{P}^{-1}(M) = \{y \in \mathbf{R}^m : \exists u : A(Py + p) + Bu \le c\}.$$

Note that rules for intersection, taking direct products and taking inverse images, as applied to polyhedral *descriptions* of operands, lead to polyhedral descriptions of the results. In contrast to this, the rules for taking sums with coefficients and images under affine mappings heavily exploit the notion of polyhedral *representation*: even when the operands in these rules are given by polyhedral descriptions, there are no simple ways to point out polyhedral *descriptions* of the results.

Exercise 1.10 *Justify the above calculus rules.*

Finally, we note that the problem of minimizing a linear form $c^T x$ over a set M given by polyhedral representation:
$$M = \{x \in \mathbf{R}^n : \exists u \in \mathbf{R}^k : Ax + Bu \le c\}$$
can be immediately reduced to an explicit LP program, namely,
$$\min_{x,u} \left\{c^T x : Ax + Bu \le c\right\}.$$

A reader with some experience in Linear Programming definitely used a lot the above "calculus of polyhedral representations" when building LPs (perhaps without clear understanding of what in fact is going on, same as Molière's Monsieur Jourdain all his life has been speaking prose without knowing it).

1.2.5 General Theorem on Alternative and Linear Programming Duality

1.2.5.1 Homogeneous Farkas Lemma

Let $a_1, ..., a_N$ be vectors from $\mathbf{R}^n$, and let a be another vector. Here we address the question: when a belongs to the cone spanned by the vectors $a_1, ..., a_N$, i.e., when a can be represented as a linear combination of a_i with *nonnegative* coefficients? A *necessary* condition is evident: if
$$a = \sum_{i=1}^n \lambda_i a_i \quad [\lambda_i \ge 0,\ i = 1, ..., N]$$

then every vector h which has nonnegative inner products with all a_i should also have nonnegative inner product with a:

$$a = \sum_i \lambda_i a_i \ \& \ \lambda_i \geq 0 \, \forall i \ \& \ h^T a_i \geq 0 \, \forall i \Rightarrow h^T a \geq 0.$$

The Homogeneous Farkas Lemma says that this evident necessary condition is also sufficient:

Lemma 1.2.1 [Homogeneous Farkas Lemma] *Let $a, a_1, ..., a_N$ be vectors from $\mathbf{R}^n$. The vector a is a conic combination of the vectors a_i (linear combination with nonnegative coefficients) if and only if every vector h satisfying $h^T a_i \geq 0$, $i = 1, ..., N$, satisfies also $h^T a \geq 0$. In other words, a homogeneous linear inequality*

$$a^T h \geq 0$$

in variable h is consequence of the system

$$a_i^T h \geq 0, \ 1 \leq i \leq N$$

of homogeneous linear inequalities if and only if it can be obtained from the inequalities of the system by "admissible linear aggregation" – taking their weighted sum with nonnegative weights.

Proof. The necessity – the "only if" part of the statement – was proved before the Farkas Lemma was formulated. Let us prove the "if" part of the Lemma. Thus, assume that every vector h satisfying $h^T a_i \geq 0 \ \forall i$ satisfies also $h^T a \geq 0$, and let us prove that a is a conic combination of the vectors a_i.

An "intelligent" proof goes as follows. The set $\text{Cone}\{a_1, ..., a_N\}$ of all conic combinations of $a_1, ..., a_N$ is polyhedrally representable (Example 3 in section 1.2.5.A.1) and as such is polyhedral (Theorem 1.2.5):

$$\text{Cone}\{a_1, ..., a_N\} = \{x \in \mathbf{R}^n : p_j^T x \geq b_j, \ 1 \leq j \leq J\}. \qquad (!)$$

Observing that $0 \in \text{Cone}\{a_1, ..., a_N\}$, we conclude that $b_j \leq 0$ for all j; and since $\lambda a_i \in \text{Cone}\{a_1, ..., a_N\}$ for every i and every $\lambda \geq 0$, we should have $\lambda p_j^T a_i \geq b_j$ for all i, j and all $\lambda \geq 0$, whence $p_j^T a_i \geq 0$ for all i and j. For every j, relation $p_j^T a_i \geq 0$ for all i implies, by the premise of the statement we want to prove, that $p_j^T a \geq 0$, and since $b_j \leq 0$, we see that $p_j^T a \geq b_j$ for all j, meaning that a indeed belongs to $\text{Cone}\{a_1, ..., a_N\}$ due to (!). ∎

An interested reader can get a better understanding of the power of Fourier-Motzkin elimination, which ultimately is the basis for the above intelligent proof, by comparing this proof with the one based on Helley's Theorem.

Proof based on Helley's Theorem. As above, we assume that every vector h satisfying $h^T a_i \geq 0 \ \forall i$ satisfies also $h^T a \geq 0$, and we want to prove that a is a conic combination of the vectors a_i.

There is nothing to prove when $a = 0$ – the zero vector of course is a conic combination of the vectors a_i. Thus, from now on we assume that $a \neq 0$.

1^0. Let

$$\Pi = \{h : a^T h = -1\},$$

and let

$$A_i = \{h \in \Pi : a_i^T h \geq 0\}.$$

Π is a hyperplane in $\mathbf{R}^n$, and every A_i is a polyhedral set contained in this hyperplane and is therefore convex.

2^0. What we know is that the intersection of all the sets A_i, $i = 1, ..., N$, is empty (since a vector h from the intersection would have nonnegative inner products with all a_i and the inner product -1 with a, and we are given that no such h exists). Let us choose the smallest, in the number of elements, of those sub-families of the family of sets $A_1, ..., A_N$ which still have empty intersection of their members; without loss of generality we may assume that this is the family $A_1, ..., A_k$. Thus, the intersection of all k sets $A_1, ..., A_k$ is empty, but the intersection of every $k-1$ sets from the family $A_1, ..., A_k$ is nonempty.

3^0. We claim that

(A) $a \in \mathrm{Lin}(\{a_1, ..., a_k\})$;

(B) The vectors $a_1, ..., a_k$ are linearly independent.

(A) is easy: assuming that $a \notin E = \mathrm{Lin}(\{a_1, ..., a_k\})$, we conclude that the orthogonal projection f of the vector a onto the orthogonal complement $E^\perp$ of E is nonzero. The inner product of f and a is the same as $f^T f$, is.e., is positive, while $f^T a_i = 0$, $i = 1, ..., k$. Taking $h = -(f^T f)^{-1} f$, we see that $h^T a = -1$ and $h^T a_i = 0$, $i = 1, ..., k$. In other words, h belongs to every set A_i, $i = 1, ..., k$, by definition of these sets, and therefore the intersection of the sets $A_1, ..., A_k$ is nonempty, which is a contradiction.

(B) is given by the Helley Theorem I. (B) is evident when $k = 1$, since in this case linear dependence of $a_!, ..., a_k$ would mean that $a_1 = 0$; by (A), this implies that $a = 0$, which is not the case. Now let us prove (B) in the case of $k > 1$. Assume, on the contrary to what should be proven, that $a_1, ..., a_k$ are linearly dependent, so that the dimension of $E = \mathrm{Lin}(\{a_1, ..., a_k\})$ is certain $m < k$. We already know from A. that $a \in E$. Now let $A'_i = A_i \cap E$. We claim that every $k - 1$ of the sets A'_i have a nonempty intersection, while all k these sets have empty intersection. The second claim is evident – since the sets $A_1, ..., A_k$ have empty intersection, the same is the case with their parts A'_i. The first claim also is easily supported: let us take $k - 1$ of the dashed sets, say, $A'_1, ..., A'_{k-1}$. By construction, the intersection of $A_1, ..., A_{k-1}$ is nonempty; let h be a vector from this intersection, i.e., a vector with nonnegative inner products with $a_1, ..., a_{k-1}$ and the product -1 with a. When replacing h with its orthogonal projection h' on E, we do not vary all these inner products, since these are products with vectors from E; thus, h' also is a common point of $A_1, ..., A_{k-1}$, and since this is a point from E, it is a common point of the dashed sets $A'_1, ..., A'_{k-1}$ as well.

Now we can complete the proof of (B): the sets $A'_1, ..., A'_k$ are convex sets belonging to the *hyperplane* $\Pi' = \Pi \cap E = \{h \in E : a^T h = -1\}$ (Π' indeed is a hyperplane in E, since $0 \neq a \in E$) *in the m-dimensional linear subspace E*. Π' is an affine subspace of the affine dimension $\ell = \dim E - 1 = m - 1 < k - 1$ (recall that we are in the situation when $m = \dim E < k$), and every $\ell + 1 \leq k - 1$ subsets from the family A'_1,...,A'_k have a nonempty intersection. From the Helley Theorem I (see Exercise 1.9) it follows that all the sets $A'_1, ..., A'_k$ have a point in common, which, as we know, is not the case. The contradiction we have got proves that $a_1, ..., a_k$ are linearly independent.

4^0. With (A) and (B) in our disposal, we can easily complete the proof of the "if" part of the Farkas Lemma. Specifically, by (A), we have

$$a = \sum_{i=1}^{k} \lambda_i a_i$$

with some real coefficients λ_i, and all we need is to prove that these coefficients are nonnegative. Assume, on the contrary, that, say, $\lambda_1 < 0$. Let us extend the (linearly independent in view of (B)) system of vectors $a_1, ..., a_k$ by vectors $f_1, ..., f_{n-k}$ to a basis in $\mathbf{R}^n$, and let $\xi_i(x)$ be the coordinates of a vector x in this basis. The function $\xi_1(x)$ is a linear form of x and therefore is the inner product with certain vector:

$$\xi_1(x) = f^T x \quad \forall x.$$

Now we have

$$f^T a = \xi_1(a) = \lambda_1 < 0$$

and

$$f^T a_i = \begin{cases} 1, & i = 1 \\ 0, & i = 2, ..., k \end{cases}$$

so that $f^T a_i \geq 0$, $i = 1, ..., k$. We conclude that a proper normalization of f – namely, the vector $|\lambda_1|^{-1} f$ – belongs to $A_1, ..., A_k$, which is the desired contradiction – by construction, this intersection is empty. ∎

1.2.5.2 General Theorem on Alternative

1.2.5.2.1 Certificates for solvability and insolvability. Consider a (finite) system of scalar inequalities with n unknowns. To be as general as possible, we do not assume for the time being the inequalities to be linear, and we allow for both non-strict and strict inequalities in the system, as well as for equalities. Since an equality can be represented by a pair of non-strict inequalities, our system can always be written as

$$f_i(x)\ \Omega_i\ 0,\ i = 1, ..., m, \tag{$\mathcal{S}$}$$

where every Ω_i is either the relation " $>$ " or the relation " $\geq$ ".

The basic question about $(\mathcal{S})$ is

(?) Whether $(\mathcal{S})$ has a solution or not.

Knowing how to answer the question (?), we are able to answer many other questions. For example, to verify whether a given real a is a lower bound on the optimal value c^* of (LP) is the same as to verify whether the system

$$\left\{\begin{array}{rcl} -c^T x + a & > & 0 \\ Ax - b & \geq & 0 \end{array}\right.$$

has no solutions.

The general question above is too difficult, and it makes sense to pass from it to a seemingly simpler one:

(??) How to certify that $(\mathcal{S})$ has, or does not have, a solution.

Imagine that you are very smart and know the correct answer to (?); how could you convince me that your answer is correct? What could be an "evident for everybody" validity certificate for your answer?

If your claim is that $(\mathcal{S})$ is solvable, a certificate could be just to point out a solution x^* to $(\mathcal{S})$. Given this certificate, one can substitute x^* into the system and check whether x^* indeed is a solution.

Assume now that your claim is that $(\mathcal{S})$ has no solutions. What could be a "simple certificate" of this claim? How one could certify a *negative* statement? This is a highly nontrivial problem not just for mathematics; for example, in criminal law: how should someone accused in a murder prove his innocence? The "real life" answer to the question "how to certify a negative statement" is discouraging: such a statement normally *cannot* be certified (this is where the rule "a person is presumed innocent until proven guilty" comes from). In mathematics, however, the situation is different: in some cases there exist "simple certificates" of negative statements. For example, in order to certify that $(\mathcal{S})$ has no solutions, it suffices to demonstrate that a consequence of $(\mathcal{S})$ is a contradictory inequality such as

$$-1 \geq 0.$$

For example, assume that λ_i, $i = 1, ..., m$, are nonnegative weights. Combining inequalities from $(\mathcal{S})$ with these weights, we come to the inequality

$$\sum_{i=1}^{m} \lambda_i f_i(x)\ \Omega\ 0 \tag{Comb(λ)}$$

where Ω is either " $>$ " (this is the case when the weight of at least one strict inequality from $(\mathcal{S})$ is positive), or " $\geq$ " (otherwise). Since the resulting inequality, due to its origin, is a consequence of the system $(\mathcal{S})$, i.e., it is satisfied by every solution to $(\mathcal{S})$, it follows that if (Comb(λ)) has no solutions at all, we can be sure that $(\mathcal{S})$ has no solution. Whenever this is the case, we may treat the corresponding vector λ as a "simple certificate" of the fact that $(\mathcal{S})$ is infeasible.

Let us look what does the outlined approach mean when $(\mathcal{S})$ is comprised of *linear* inequalities:

$$(\mathcal{S}):\quad \{a_i^T x\ \Omega_i\ b_i,\ i = 1, ..., m\} \quad \left[\Omega_i = \left\{\begin{array}{l} " > " \\ " \geq " \end{array}\right.\right]$$

Here the "combined inequality" is linear as well:

$$(\mathrm{Comb}(\lambda)): \qquad (\sum_{i=1}^m \lambda a_i)^T x \ \Omega \ \sum_{i=1}^m \lambda b_i$$

(Ω is " $>$ " whenever $\lambda_i > 0$ for at least one i with $\Omega_i =$ " $>$ ", and Ω is " $\geq$ " otherwise). Now, when can a *linear* inequality

$$d^T x \ \Omega \ e$$

be contradictory? Of course, it can happen only when $d = 0$. Whether in this case the inequality is contradictory, it depends on what is the relation Ω: if $\Omega =$ " $>$ ", then the inequality is contradictory if and only if $e \geq 0$, and if $\Omega =$ " $\geq$ ", it is contradictory if and only if $e > 0$. We have established the following simple result:

Proposition 1.2.1 *Consider a system of linear inequalities*

$$(\mathcal{S}): \qquad \left\{ \begin{array}{rcl} a_i^T x & > & b_i, \ i = 1, ..., m_{\mathrm{s}}, \\ a_i^T x & \geq & b_i, \ i = m_{\mathrm{s}} + 1, ..., m. \end{array} \right.$$

with n-dimensional vector of unknowns x. Let us associate with $(\mathcal{S})$ two systems of linear inequalities and equations with m-dimensional vector of unknowns λ:

$$\mathcal{T}_{\mathrm{I}}: \qquad \left\{ \begin{array}{rcl} (a) \quad \lambda & \geq & 0; \\ (b) \quad \sum\limits_{i=1}^m \lambda_i a_i & = & 0; \\ (c_{\mathrm{I}}) \quad \sum\limits_{i=1}^m \lambda_i b_i & \geq & 0; \\ \hline (d_{\mathrm{I}}) \quad \sum\limits_{i=1}^{m_{\mathrm{s}}} \lambda_i & > & 0. \end{array} \right.$$

$$\mathcal{T}_{\mathrm{II}}: \qquad \left\{ \begin{array}{rcl} (a) \quad \lambda & \geq & 0; \\ (b) \quad \sum\limits_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\mathrm{II}}) \quad \sum\limits_{i=1}^m \lambda_i b_i & > & 0. \end{array} \right.$$

Assume that at least one of the systems $\mathcal{T}_{\mathrm{I}}$, $\mathcal{T}_{\mathrm{II}}$ is solvable. Then the system $(\mathcal{S})$ is infeasible.

1.2.5.2.2 General Theorem on Alternative. Proposition 1.2.1 says that in some cases it is easy to certify infeasibility of a linear system of inequalities: a "simple certificate" is a solution to another system of linear inequalities. Note, however, that the existence of a certificate of this latter type is to the moment only a *sufficient*, but not a *necessary*, condition for the infeasibility of $(\mathcal{S})$. A fundamental result in the theory of linear inequalities is that the sufficient condition in question is in fact also necessary:

Theorem 1.2.6 [General Theorem on Alternative] *In the notation from Proposition 1.2.1, system $(\mathcal{S})$ has no solutions if and only if either $\mathcal{T}_{\mathrm{I}}$, or $\mathcal{T}_{\mathrm{II}}$, or both these systems, are solvable.*

Proof. GTA is a more or less straightforward corollary of the Homogeneous Farkas Lemma. Indeed, in view of Proposition 1.2.1, all we need to prove is that if $(\mathcal{S})$ has no solution, then at least one of the systems $\mathcal{T}_{\mathrm{I}}$, or $\mathcal{T}_{\mathrm{II}}$ is solvable. Thus, assume that $(\mathcal{S})$ has no solutions, and let us look at the consequences. Let us associate with $(\mathcal{S})$ the following system of *homogeneous* linear inequalities in variables x, τ, ϵ:

$$\begin{array}{llrcll} (a) & & \tau & -\epsilon & \geq & 0 \\ (b) & a_i^T x & -b_i \tau & -\epsilon & \geq & 0, \ i = 1, ..., m_{\mathrm{s}} \\ (c) & a_i^T x & -b_i \tau & & \geq & 0, \ i = m_{\mathrm{s}} + 1, ..., m \end{array} \qquad (1.2.3)$$

We claim that

(!) *For every solution to* (1.2.3), *one has* $\epsilon \le 0$.

Indeed, assuming that (1.2.3) has a solution x, τ, ϵ with $\epsilon > 0$, we conclude from (1.2.3.a) that $\tau > 0$; from (1.2.3.$b-c$) it now follows that $\tau^{-1}x$ is a solution to $(\mathcal{S})$, while the latter system is unsolvable.

Now, (!) says that the homogeneous linear inequality

$$-\epsilon \ge 0 \tag{1.2.4}$$

is a consequence of the system of homogeneous linear inequalities (1.2.3). By Homogeneous Farkas Lemma, it follows that there exist nonnegative weights ν, λ_i, $i = 1, ..., m$, such that the vector of coefficients of the variables x, τ, ϵ in the left hand side of (1.2.4) is linear combination, with the coefficients $\nu, \lambda_1, ..., \lambda_m$, of the vectors of coefficients of the variables in the inequalities from (1.2.3):

$$\begin{array}{lrcl} (a) & \sum\limits_{i=1}^{m} \lambda_i a_i & = & 0 \\ (b) & -\sum\limits_{i=1}^{m} \lambda_i b_i + \nu & = & 0 \\ (c) & -\sum\limits_{i=1}^{m_s} \lambda_i - \nu & = & -1 \end{array} \tag{1.2.5}$$

Recall that by their origin, ν and all λ_i are nonnegative. Now, it may happen that $\lambda_1, ..., \lambda_{m_s}$ are zero. In this case $\nu > 0$ by (1.2.5.c), and relations (1.2.5$a-b$) say that $\lambda_1, ..., \lambda_m$ solve $\mathcal{T}_{\rm II}$. In the remaining case (that is, when not all $\lambda_1, ..., \lambda_{m_s}$ are zero, or, which is the same, when $\sum\limits_{i=1}^{m_s} \lambda_i > 0$), the same relations say that $\lambda_1, ..., \lambda_m$ solve $\mathcal{T}_{\rm I}$. ■

1.2.5.2.3 Corollaries of the Theorem on Alternative. We formulate here explicitly two very useful principles following from the Theorem on Alternative:

A. *A system of linear inequalities*

$$a_i^T x \ \Omega_i \ b_i, \ \ i = 1, ..., m$$

has no solutions if and only if one can combine the inequalities of the system in a linear fashion (i.e., multiplying the inequalities by nonnegative weights, adding the results and passing, if necessary, from an inequality $a^T x > b$ to the inequality $a^T x \ge b$) to get a contradictory inequality, namely, either the inequality $0^T x \ge 1$, or the inequality $0^T x > 0$.

B. *A linear inequality*

$$a_0^T x \ \Omega_0 \ b_0$$

is a consequence of a solvable system of linear inequalities

$$a_i^T x \ \Omega_i \ b_i, \ \ i = 1, ..., m$$

if and only if it can be obtained by combining, in a linear fashion, the inequalities of the system and the trivial inequality $0 > -1$.

It should be stressed that the above principles are highly nontrivial and very deep. Consider, e.g., the following system of 4 linear inequalities with two variables u, v:

$$\begin{array}{l} -1 \le u \le 1 \\ -1 \le v \le 1. \end{array}$$

From these inequalities it follows that

$$u^2 + v^2 \le 2, \tag{!}$$

which in turn implies, by the Cauchy inequality, the linear inequality $u+v \le 2$:

$$u+v = 1\times u + 1 \times v \le \sqrt{1^2+1^2}\sqrt{u^2+v^2} \le (\sqrt{2})^2 = 2. \qquad (!!)$$

The concluding inequality is linear and is a consequence of the original system, but in the demonstration of this fact both steps (!) and (!!) are "highly nonlinear". It is absolutely unclear a priori why the same consequence can, as it is stated by Principle **A**, be derived from the system in a linear manner as well [of course it can – it suffices just to add two inequalities $u \le 1$ and $v \le 1$].

Note that the Theorem on Alternative and its corollaries **A** and **B** heavily exploit the fact that we are speaking about *linear* inequalities. For example, consider the following 2 quadratic and 2 linear inequalities with two variables:

$$\begin{array}{lrcl} (a) & u^2 & \ge & 1; \\ (b) & v^2 & \ge & 1; \\ (c) & u & \ge & 0; \\ (d) & v & \ge & 0; \end{array}$$

along with the quadratic inequality

$$(e) \quad uv \ge 1.$$

The inequality (e) is clearly a consequence of (a) – (d). However, if we extend the system of inequalities (a) – (b) by all "trivial" (i.e., identically true) linear and quadratic inequalities with 2 variables, like $0 > -1$, $u^2+v^2 \ge 0$, $u^2+2uv+v^2 \ge 0$, $u^2-uv+v^2 \ge 0$, etc., and ask whether (e) can be derived in a *linear* fashion from the inequalities of the extended system, the answer will be negative. Thus, Principle **A** fails to be true already for quadratic inequalities (which is a great sorrow – otherwise there were no difficult problems at all!)

1.2.5.3 Application: Linear Programming Duality

We are about to use the Theorem on Alternative to obtain the basic results of the LP duality theory.

1.2.5.3.1 Dual to an LP program: the origin The motivation for constructing the problem dual to an LP program

$$c^* = \min_x \left\{ c^T x : Ax - b \ge 0 \right\} \qquad \left[A = \left[\begin{array}{c} a_1^T \\ a_2^T \\ ... \\ a_m^T \end{array} \right] \in \mathbf{R}^{m\times n} \right] \qquad (LP)$$

is the desire to generate, in a systematic way, lower bounds on the optimal value c^* of (LP). An evident way to bound from below a given function $f(x)$ in the domain given by system of inequalities

$$g_i(x) \ge b_i,\ i = 1, ..., m, \qquad (1.2.6)$$

is offered by what is called the *Lagrange duality* (to be investigated in-depth in Section 3.2) and is as follows:

Lagrange Duality:
• *Let us look at all inequalities which can be obtained from* (1.2.6) *by linear aggregation, i.e., at the inequalities of the form*

$$\sum_i y_i g_i(x) \ge \sum_i y_i b_i \qquad (1.2.7)$$

with the "aggregation weights" $y_i \geq 0$. Note that the inequality (1.2.7)*, due to its origin, is valid on the entire set X of solutions of* (1.2.6).

• *Depending on the choice of aggregation weights, it may happen that the left hand side in* (1.2.7) *is $\leq f(x)$ for all $x \in \mathbf{R}^n$. Whenever it is the case, the right hand side $\sum_i y_i b_i$ of* (1.2.7) *is a lower bound on f in X.*

Indeed, on X the quantity $\sum_i y_i b_i$ is a lower bound on $y_i g_i(x)$, and for y in question the latter function of x is everywhere $\leq f(x)$.

It follows that

• *The optimal value in the problem*

$$\max_y \left\{ \sum_i y_i b_i : \begin{array}{cc} y \geq 0, & (a) \\ \sum_i y_i g_i(x) \leq f(x) \ \forall x \in \mathbf{R}^n & (b) \end{array} \right\} \tag{1.2.8}$$

is a lower bound on the values of f on the set of solutions to the system (1.2.6).

Let us look what happens with the Lagrange duality when f and g_i are homogeneous linear functions: $f = c^T x$, $g_i(x) = a_i^T x$. In this case, the requirement (1.2.8.b) merely says that $c = \sum_i y_i a_i$ (or, which is the same, $A^T y = c$ due to the origin of A). Thus, problem (1.2.8) becomes the Linear Programming problem

$$\max_y \left\{ b^T y : A^T y = c, \, y \geq 0 \right\}, \tag{LP*}$$

which is nothing but the LP dual of (LP).

By the construction of the dual problem,

[Weak Duality] *The optimal value in* (LP*) *is less than or equal to the optimal value in* (LP).

In fact, the "less than or equal to" in the latter statement is "equal", provided that the optimal value c^* in (LP) is a number (i.e., (LP) is feasible and below bounded). To see that this indeed is the case, note that a real a is a lower bound on c^* if and only if $c^T x \geq a$ whenever $Ax \geq b$, or, which is the same, if and only if the system of linear inequalities

$$(\mathcal{S}_a): \qquad -c^T x > -a, Ax \geq b$$

has no solution. We know by the Theorem on Alternative that the latter fact means that some other system of linear equalities (more exactly, at least one of a certain pair of systems) does have a solution. More precisely,

(*) ($\mathcal{S}_a$) *has no solutions if and only if at least one of the following two systems with $m+1$ unknowns:*

$$\mathcal{T}_{\rm I}: \quad \left\{ \begin{array}{lrcl} (a) & \lambda = (\lambda_0, \lambda_1, ..., \lambda_m) & \geq & 0; \\ (b) & -\lambda_0 c + \sum_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\rm I}) & -\lambda_0 a + \sum_{i=1}^m \lambda_i b_i & \geq & 0; \\ (d_{\rm I}) & \lambda_0 & > & 0, \end{array} \right.$$

or

$$\mathcal{T}_{\rm II}: \quad \left\{ \begin{array}{lrcl} (a) & \lambda = (\lambda_0, \lambda_1, ..., \lambda_m) & \geq & 0; \\ (b) & -\lambda_0 c - \sum_{i=1}^m \lambda_i a_i & = & 0; \\ \hline (c_{\rm II}) & -\lambda_0 a - \sum_{i=1}^m \lambda_i b_i & > & 0 \end{array} \right.$$

– *has a solution.*

Now assume that (LP) *is feasible*. We claim that *under this assumption* $(\mathcal{S}_a)$ *has no solutions if and only if* $\mathcal{T}_{\rm I}$ *has a solution*.

> The implication "$\mathcal{T}_{\rm I}$ has a solution $\Rightarrow$ $(\mathcal{S}_a)$ has no solution" is readily given by the above remarks. To verify the inverse implication, assume that $(\mathcal{S}_a)$ has no solutions and the system $Ax \le b$ has a solution, and let us prove that then $\mathcal{T}_{\rm I}$ has a solution. If $\mathcal{T}_{\rm I}$ has no solution, then by (*) $\mathcal{T}_{\rm II}$ has a solution and, moreover, $\lambda_0 = 0$ for (every) solution to $\mathcal{T}_{\rm II}$ (since a solution to the latter system with $\lambda_0 > 0$ solves $\mathcal{T}_{\rm I}$ as well). But the fact that $\mathcal{T}_{\rm II}$ has a solution λ with $\lambda_0 = 0$ is independent of the values of a and c; if this fact would take place, it would mean, by the same Theorem on Alternative, that, e.g., the following instance of $(\mathcal{S}_a)$:
>
> $$0^T x \ge -1, Ax \ge b$$
>
> has no solutions. The latter means that the system $Ax \ge b$ has no solutions – a contradiction with the assumption that (LP) is feasible. ∎

Now, if $\mathcal{T}_{\rm I}$ has a solution, this system has a solution with $\lambda_0 = 1$ as well (to see this, pass from a solution λ to the one λ/λ_0; this construction is well-defined, since $\lambda_0 > 0$ for every solution to $\mathcal{T}_{\rm I}$). Now, an $(m+1)$-dimensional vector $\lambda = (1, y)$ is a solution to $\mathcal{T}_{\rm I}$ if and only if the m-dimensional vector y solves the system of linear inequalities and equations

$$\begin{array}{rcl} y & \ge & 0; \\ A^T y \equiv \sum\limits_{i=1}^m y_i a_i & = & c; \\ b^T y & \ge & a \end{array} \qquad \text{(D)}$$

Summarizing our observations, we come to the following result.

Proposition 1.2.2 *Assume that system* (D) *associated with the LP program* (LP) *has a solution* (y, a). *Then* a *is a lower bound on the optimal value in* (LP). *Vice versa, if* (LP) *is feasible and* a *is a lower bound on the optimal value of* (LP), *then* a *can be extended by a properly chosen* m*-dimensional vector* y *to a solution to* (D).

We see that the entity responsible for lower bounds on the optimal value of (LP) is the system (D): every solution to the latter system induces a bound of this type, and *in the case when* (LP) *is feasible*, all lower bounds can be obtained from solutions to (D). Now note that if (y, a) is a solution to (D), then the pair $(y, b^T y)$ also is a solution to the same system, and the lower bound $b^T y$ on c^* is not worse than the lower bound a. Thus, as far as lower bounds on c^* are concerned, we lose nothing by restricting ourselves to the solutions (y, a) of (D) with $a = b^T y$; the best lower bound on c^* given by (D) is therefore the optimal value of the problem $\max_y \left\{b^T y : A^T y = c, y \ge 0\right\}$, which is nothing but the dual to (LP) problem (LP*). Note that (LP*) is also a Linear Programming program.

All we know about the dual problem to the moment is the following:

Proposition 1.2.3 *Whenever* y *is a feasible solution to* (LP*), *the corresponding value of the dual objective* $b^T y$ *is a lower bound on the optimal value* c^* *in* (LP). *If* (LP) *is feasible, then for every* $a \le c^*$ *there exists a feasible solution* y *of* (LP*) *with* $b^T y \ge a$.

1.2.5.3.2 Linear Programming Duality Theorem. Proposition 1.2.3 is in fact equivalent to the following

Theorem 1.2.7 [Duality Theorem in Linear Programming] *Consider a linear programming program*

$$\min_x \left\{c^T x : Ax \ge b\right\} \qquad \text{(LP)}$$

along with its dual

$$\max_y \left\{b^T y : A^T y = c, y \ge 0\right\} \qquad \text{(LP}^*\text{)}$$

Then

1) *The duality is symmetric: the problem dual to dual is equivalent to the primal;*

2) *The value of the dual objective at every dual feasible solution is $\leq$ the value of the primal objective at every primal feasible solution*

3) *The following 5 properties are equivalent to each other:*

(i) *The primal is feasible and bounded below.*

(ii) *The dual is feasible and bounded above.*

(iii) *The primal is solvable.*

(iv) *The dual is solvable.*

(v) *Both primal and dual are feasible.*

Whenever (i) $\equiv$ (ii) $\equiv$ (iii) $\equiv$ (iv) $\equiv$ (v) *is the case, the optimal values of the primal and the dual problems are equal to each other.*

Proof. 1) is quite straightforward: writing the dual problem (LP*) in our standard form, we get

$$\min_y \left\{ -b^T y : \begin{bmatrix} I_m \\ A^T \\ -A^T \end{bmatrix} y - \begin{pmatrix} 0 \\ -c \\ c \end{pmatrix} \geq 0 \right\},$$

where I_m is the m-dimensional unit matrix. Applying the duality transformation to the latter problem, we come to the problem

$$\max_{\xi,\eta,\zeta} \left\{ 0^T\xi + c^T\eta + (-c)^T\zeta : \begin{array}{rcl} \xi & \geq & 0 \\ \eta & \geq & 0 \\ \zeta & \geq & 0 \\ \xi - A\eta + A\zeta & = & -b \end{array} \right\},$$

which is clearly equivalent to (LP) (set $x = \eta - \zeta$).

2) is readily given by Proposition 1.2.3.

3):

(i)$\Rightarrow$(iv): If the primal is feasible and bounded below, its optimal value c^* (which of course is a lower bound on itself) can, by Proposition 1.2.3, be (non-strictly) majorized by a quantity $b^T y^*$, where y^* is a feasible solution to (LP*). In the situation in question, of course, $b^T y^* = c^*$ (by already proved item 2)); on the other hand, in view of the same Proposition 1.2.3, the optimal value in the dual is $\leq c^*$. We conclude that the optimal value in the dual is attained and is equal to the optimal value in the primal.

(iv)$\Rightarrow$(ii): evident;

(ii)$\Rightarrow$(iii): This implication, in view of the primal-dual symmetry, follows from the implication (i)$\Rightarrow$(iv).

(iii)$\Rightarrow$(i): evident.

We have seen that (i)$\equiv$(ii)$\equiv$(iii)$\equiv$(iv) and that the first (and consequently each) of these 4 equivalent properties implies that the optimal value in the primal problem is equal to the optimal value in the dual one. All which remains is to prove the equivalence between (i)–(iv), on one hand, and (v), on the other hand. This is immediate: (i)–(iv), of course, imply (v); vice versa, in the case of (v) the primal is not only feasible, but also bounded below (this is an immediate consequence of the feasibility of the dual problem, see 2)), and (i) follows. ∎

An immediate corollary of the LP Duality Theorem is the following *necessary and sufficient* optimality condition in LP:

Theorem 1.2.8 [Necessary and sufficient optimality conditions in linear programming] *Consider an LP program* (LP) *along with its dual* (LP*). *A pair (x, y) of primal and dual feasible solutions is comprised of optimal solutions to the respective problems if and only if*

$$y_i[Ax - b]_i = 0,\ i = 1, ..., m, \qquad \text{[complementary slackness]}$$

likewise as if and only if

$$c^T x - b^T y = 0 \qquad \text{[zero duality gap]}$$

Indeed, the "zero duality gap" optimality condition is an immediate consequence of the fact that the value of primal objective at every primal feasible solution is $\geq$ the value of the dual objective at every dual feasible solution, while the optimal values in the primal and the dual are equal to each other, see Theorem 1.2.7. The equivalence between the "zero duality gap" and the "complementary slackness" optimality conditions is given by the following computation: whenever x is primal feasible and y is dual feasible, the products $y_i[Ax - b]_i$, $i = 1, ..., m$, are nonnegative, while the sum of these products is precisely the duality gap:

$$y^T[Ax - b] = (A^T y)^T x - b^T y = c^T x - b^T y.$$

Thus, the duality gap can vanish at a primal-dual feasible pair (x, y) if and only if all products $y_i[Ax - b]_i$ for this pair are zeros.

1.2.6 Separation Theorem

1.2.6.1 Separation: definition

Recall that a *hyperplane* M in $\mathbf{R}^n$ is, by definition, an affine subspace of the dimension $n - 1$. By Proposition A.3.7, hyperplanes are exactly the same as level sets of nontrivial linear forms:

$$\begin{array}{c} M \subset \mathbf{R}^n \text{ is a hyperplane} \\ \Updownarrow \\ \exists a \in \mathbf{R}^n, b \in \mathbf{R}, a \neq 0 : \quad M = \{x \in \mathbf{R}^n : a^T x = b\} \end{array}$$

We can, consequently, associate with the hyperplane (or, better to say, with the associated linear form a; this form is defined uniquely, up to multiplication by a nonzero real) the following sets:

- "upper" and "lower" open half-spaces $M^{++} = \{x \in \mathbf{R}^n : a^T x > b\}$, $M^{--} = \{x \in \mathbf{R}^n : a^T x < b\}$; these sets clearly are convex, and since a linear form is continuous, and the sets are given by strict inequalities on the value of a continuous function, they indeed are open.

  Note that since a is uniquely defined by M, up to multiplication by a nonzero real, these open half-spaces are uniquely defined by the hyperplane, up to swapping the "upper" and the "lower" ones (which half-space is "upper", it depends on the particular choice of a);
- "upper" and "lower" closed half-spaces $M^+ = \{x \in \mathbf{R}^n : a^T x \geq b\}$, $M^- = \{x \in \mathbf{R}^n : a^T x \leq b\}$; these are also convex sets, now closed (since they are given by non-strict inequalities on the value of a continuous function). It is easily seen that the closed upper/lower half-space is the closure of the corresponding open half-space, and M itself is the boundary (i.e., the complement of the interior to the closure) of all four half-spaces.

It is clear that our half-spaces and M itself partition $\mathbf{R}^n$:

$$\mathbf{R}^n = M^{--} \cup M \cup M^{++}$$

(partitioning by disjoint sets),

$$\mathbf{R}^n = M^- \cup M^+$$

(M is the intersection of the right hand side sets).

Now we define the basic notion of *separation* of two convex sets T and S by a hyperplane.

Definition 1.2.2 [separation] *Let S, T be two nonempty convex sets in $\mathbf{R}^n$.*

- *A hyperplane*
$$M = \{x \in \mathbf{R}^n : a^T x = b\} \quad [a \neq 0]$$
is said to separate S and T, if, first,
$$S \subset \{x : a^T x \leq b\}, \quad T \subset \{x : a^T x \geq b\}$$
(i.e., S and T belong to the opposite closed half-spaces into which M splits $\mathbf{R}^n$), and, second, at least one of the sets S, T is not contained in M itself:
$$S \cup T \not\subset M.$$
The separation is called strong, if there exist b', b'', $b' < b < b''$, such that
$$S \subset \{x : a^T x \leq b'\}, \quad T \subset \{x : a^T x \geq b''\}.$$
- *A linear form $a \neq 0$ is said to separate (strongly separate) S and T, if for properly chosen b the hyperplane $\{x : a^T x = b\}$ separates (strongly separates) S and T.*
- *We say that S and T can be (strongly) separated, if there exists a hyperplane which (strongly) separates S and T.*

For example,

- the hyperplane $\{x : a^T x \equiv x_2 - x_1 = 1\}$ in $\mathbf{R}^2$ strongly separates convex polyhedral sets $T = \{x \in \mathbf{R}^2 : 0 \leq x_1 \leq 1, 3 \leq x_2 \leq 5\}$ and $S = \{x \in \mathbf{R}^2 : x_2 = 0; x_1 \geq -1\}$;
- the hyperplane $\{x : a^T x \equiv x = 1\}$ in $\mathbf{R}^1$ separates (but not strongly separates) the convex sets $S = \{x \leq 1\}$ and $T = \{x \geq 1\}$;
- the hyperplane $\{x : a^T x \equiv x_1 = 0\}$ in $\mathbf{R}^2$ separates (but not strongly separates) the sets $S = \{x \in \mathbf{R}^2 :, x_1 < 0, x_2 \geq -1/x_1\}$ and $T = \{x \in \mathbf{R}^2 : x_1 > 0, x_2 > 1/x_1\}$;
- the hyperplane $\{x : a^T x \equiv x_2 - x_1 = 1\}$ in $\mathbf{R}^2$ does *not* separate the convex sets $S = \{x \in \mathbf{R}^2 : x_2 \geq 1\}$ and $T = \{x \in \mathbf{R}^2 : x_2 = 0\}$;
- the hyperplane $\{x : a^T x \equiv x_2 = 0\}$ in $\mathbf{R}^2$ does not separate the sets $S = \{x \in \mathbf{R}^2 : x_2 = 0, x_1 \leq -1\}$ and $T = \{x \in \mathbf{R}^2 : x_2 = 0, x_1 \geq 1\}$.

The following Exercise presents an equivalent description of separation:

Exercise 1.11 *Let S, T be nonempty convex sets in $\mathbf{R}^n$. Prove that a linear form a separates S and T if and only if*
$$\sup_{x \in S} a^T x \leq \inf_{y \in T} a^T y$$
and
$$\inf_{x \in S} a^T x < \sup_{y \in T} a^T y.$$
This separation is strong if and only if
$$\sup_{x \in S} a^T x < \inf_{y \in T} a^T y.$$

Exercise 1.12 *Whether the sets $S = \{x \in \mathbf{R}^2 : x_1 > 0, x_2 \geq 1/x_1\}$ and $T = \{x \in \mathbf{R}^2 : x_1 < 0, x_2 \geq -1/x_1\}$ can be separated? Whether they can be strongly separated?*

1.2.6.2 Separation Theorem

Theorem 1.2.9 [Separation Theorem] *Let S and T be nonempty convex sets in $\mathbf{R}^n$.*
(i) *S and T can be separated if and only if their relative interiors do not intersect:* $\mathrm{rint}\, S \cap \mathrm{rint}\, T = \emptyset$.
(ii) *S and T can be strongly separated if and only if the sets are at a positive distance from each other:*

$$\mathrm{dist}(S,T) \equiv \inf\{\| x-y \|_2 : x \in S, y \in T\} > 0.$$

In particular, if S, T are closed nonempty non-intersecting convex sets and one of these sets is compact, S and T can be strongly separated.

Proof takes several steps.

(i), Necessity. Assume that S, T can be separated, so that for certain $a \neq 0$ we have

$$\inf_{x\in S} a^T x \leq \inf_{y\in T} a^T y; \quad \inf_{x\in S} a^T x < \sup_{y\in T} a^T y. \tag{1.2.9}$$

We should lead to a contradiction the assumption that $\mathrm{rint}\, S$ and $\mathrm{rint}\, T$ have in common certain point $\bar{x}$. Assume that it is the case; then from the first inequality in (1.2.9) it is clear that $\bar{x}$ maximizes the linear function $f(x) = a^T x$ on S and simultaneously minimizes this function on T. Now, we have the following simple and important

> **Lemma 1.2.2** *A linear function $f(x) = a^T x$ can attain its maximum/minimum over a convex set Q at a point $x \in \mathrm{rint}\, Q$ if and only if the function is constant on Q.*
>
> **Proof.** "if" part is evident. To prove the "only if" part, let $\bar{x} \in \mathrm{rint}\, Q$ be, say, a minimizer of f over Q and y be an arbitrary point of Q; we should prove that $f(\bar{x}) = f(y)$. There is nothing to prove if $y = \bar{x}$, so let us assume that $y \neq \bar{x}$. Since $\bar{x} \in \mathrm{rint}\, Q$, the segment $[y, \bar{x}]$, which is contained in M, can be extended a little bit through the point $\bar{x}$, not leaving M (since $\bar{x} \in \mathrm{rint}\, Q$), so that there exists $z \in Q$ such that $\bar{x} \in [y, z)$, i.e., $\bar{x} = (1-\lambda)y + \lambda z$ with certain $\lambda \in (0, 1]$; since $y \neq \bar{x}$, we have in fact $\lambda \in (0, 1)$. Since f is linear, we have
>
> $$f(\bar{x}) = (1-\lambda) f(y) + \lambda f(z);$$
>
> since $f(\bar{x}) \leq \min\{f(y), f(z)\}$ and $0 < \lambda < 1$, this relation can be satisfied only when $f(\bar{x}) = f(y) = f(z)$. □

By Lemma 1.2.2, $f(x) = f(\bar{x})$ on S and on T, so that $f(\cdot)$ is constant on $S \cup T$, which yields the desired contradiction with the second inequality in (1.2.9). □

(i), Sufficiency. The proof of sufficiency part of the Separation Theorem is much more instructive. There are several ways to prove it, and we choose the one which goes via the Homogeneous Farkas Lemma 1.2.1, which is extremely important in its own right.

(i), Sufficiency, Step 1: Separation of a convex polytope and a point outside the polytope. Let us start with seemingly very particular case of the Separation Theorem – the one where S is the convex full points $x_1, ..., x_N$, and T is a singleton $T = \{x\}$ which does not belong to S. We intend to prove that in this case there exists a linear form which separates x and S; in fact we shall prove even the existence of strong separation.

Let us associate with n-dimensional vectors $x_1, ..., x_N, x$ the $(n+1)$-dimensional vectors $a = \begin{pmatrix} x \\ 1 \end{pmatrix}$ and $a_i = \begin{pmatrix} x_i \\ 1 \end{pmatrix}$, $i = 1, ..., N$. We claim that a does not belong to the conic hull of $a_1, ..., a_N$. Indeed, if a

would be representable as a linear combination of $a_1, ..., a_N$ with nonnegative coefficients, then, looking at the last, $(n+1)$-st, coordinates in such a representation, we would conclude that the sum of coefficients should be 1, so that the representation, actually, represents x as a convex combination of $x_1, ..., x_N$, which was assumed to be impossible.

Since a does not belong to the conic hull of $a_1, ..., a_N$, by the Homogeneous Farkas Lemma (Lemma 1.2.1) there exists a vector $h = \begin{pmatrix} f \\ \alpha \end{pmatrix} \in \mathbf{R}^{n+1}$ which "separates" a and $a_1, ..., a_N$ in the sense that

$$h^T a > 0, \ h^T a_i \leq 0, \ i = 1, ..., N,$$

whence, of course,

$$h^T a > \max_i h^T a_i.$$

Since the components in all the inner products $h^T a, h^T a_i$ coming from the $(n+1)$-st coordinates are equal to each other , we conclude that the n-dimensional component f of h separates x and $x_1, ..., x_N$:

$$f^T x > \max_i f^T x_i.$$

Since for every convex combination $y = \sum_i \lambda_i x_i$ of the points x_i one clearly has $f^T y \leq \max_i f^T x_i$, we conclude, finally, that

$$f^T x > \max_{y \in \mathrm{Conv}(\{x_1, ..., x_N\})} f^T y,$$

so that f strongly separates $T = \{x\}$ and $S = \mathrm{Conv}(\{x_1, ..., x_N\})$. □

(i), Sufficiency, Step 2: Separation of a convex set and a point outside of the set. Now consider the case when S is an arbitrary nonempty convex set and $T = \{x\}$ is a singleton outside S (the difference with Step 1 is that now S is not assumed to be a polytope).

First of all, without loss of generality we may assume that S contains 0 (if it is not the case, we may subject S and T to translation $S \mapsto p + S$, $T \mapsto p + T$ with $p \in -S$). Let L be the linear span of S. If $x \notin L$, the separation is easy: taking as f the orthogonal to L component of x, we shall get

$$f^T x = f^T f > 0 = \max_{y \in S} f^T y,$$

so that f strongly separates S and $T = \{x\}$.

It remains to consider the case when $x \in L$. Since $S \subset L$, $x \in L$ and $x \notin S$, L is a nonzero linear subspace; w.l.o.g., we can assume that $L = \mathbf{R}^n$.

Let $\Sigma = \{h : \| h \|_2 = 1\}$ be the unit sphere in $L = \mathbf{R}^n$. This is a closed and bounded set in $\mathbf{R}^n$ (boundedness is evident, and closedness follows from the fact that $\| \cdot \|_2$ is continuous). Consequently, Σ is a compact set. Let us prove that there exists $f \in \Sigma$ which separates x and S in the sense that

$$f^T x \geq \sup_{y \in S} f^T y. \tag{1.2.10}$$

Assume, on the contrary, that no such f exists, and let us lead this assumption to a contradiction. Under our assumption for every $h \in \Sigma$ there exists $y_h \in S$ such that

$$h^T y_h > h^T x.$$

Since the inequality is strict, it immediately follows that there exists a neighbourhood U_h of the vector h such that

$$(h')^T y_h > (h')^T x \quad \forall h' \in U_h. \tag{1.2.11}$$

The family of open sets $\{U_h\}_{h \in \Sigma}$ covers Σ; since Σ is compact, we can find a finite subfamily $U_{h_1}, ..., U_{h_N}$ of the family which still covers Σ. Let us take the corresponding points $y_1 = y_{h_1}, y_2 = y_{h_2}, ..., y_N = y_{h_N}$

and the polytope $S' = \mathrm{Conv}(\{y_1, ..., y_N\})$ spanned by the points. Due to the origin of y_i, all of them are points from S; since S is convex, the polytope S' is contained in S and, consequently, does not contain x. By Step 1, x can be strongly separated from S': there exists a such that

$$a^T x > \sup_{y \in S'} a^T y. \tag{1.2.12}$$

By normalization, we may also assume that $\| a \|_2 = 1$, so that $a \in \Sigma$. Now we get a contradiction: since $a \in \Sigma$ and $U_{h_1}, ..., U_{h_N}$ form a covering of Σ, a belongs to certain U_{h_i}. By construction of U_{h_i} (see (1.2.11)), we have

$$a^T y_i \equiv a^T y_{h_i} > a^T x,$$

which contradicts (1.2.12) – recall that $y_i \in S'$.

The contradiction we get proves that there exists $f \in \Sigma$ satisfying (1.2.10). We claim that f separates S and $\{x\}$; in view of (1.2.10), all we need to verify our claim is to show that the linear form $f(y) = f^T y$ is non-constant on $S \cup T$, which is evident: we are in the situation when $0 \in S$ and $L \equiv \mathrm{Lin}(S) = \mathbf{R}^n$ and $f \neq 0$, so that $f(y)$ is non-constant already on S. □

Mathematically oriented reader should take into account that the simple-looking reasoning underlying Step 2 in fact brings us into a completely new world. Indeed, the considerations at Step 1 and in the proof of Homogeneous Farkas Lemma are "pure arithmetic" – we never used things like convergence, compactness, etc., and used rational arithmetic only – no square roots, etc. It means that the Homogeneous Farkas Lemma and the result stated a Step 1 remain valid if we, e.g., replace our universe $\mathbf{R}^n$ with the space $\mathbf{Q}^n$ of n-dimensional rational vectors (those with rational coordinates; of course, the multiplication by reals in this space should be restricted to multiplication by rationals). The "rational" Farkas Lemma or the possibility to separate a rational vector from a "rational" polytope by a rational linear form, which is the "rational" version of the result of Step 1, definitely are of interest (e.g., for Integer Programming). In contrast to these "purely arithmetic" considerations, at Step 2 we used compactness – something heavily exploiting the fact that our universe is $\mathbf{R}^n$ and not, say, $\mathbf{Q}^n$ (in the latter space bounded and closed sets not necessary are compact). Note also that we could not avoid things like compactness arguments at Step 2, since the very fact we are proving is true in $\mathbf{R}^n$ but not in $\mathbf{Q}^n$. Indeed, consider the "rational plane" – the universe comprised of all 2-dimensional vectors with rational entries, and let S be the half-plane in this rational plane given by the linear inequality

$$x_1 + \alpha x_2 \leq 0,$$

where α is irrational. S clearly is a "convex set" in $\mathbf{Q}^2$; it is immediately seen that a point outside this set cannot be separated from S by a rational linear form.

(i), Sufficiency, Step 3: Separation of two nonempty and non-intersecting convex sets. Now we are ready to prove that two nonempty and non-intersecting convex sets S and T can be separated. To this end consider the arithmetic difference

$$\Delta = S - T = \{x - y : x \in S, y \in T\}.$$

By Proposition 1.1.6.3, Δ is convex (and, of course, nonempty) set; since S and T do not intersect, Δ does not contain 0. By Step 2, we can separate Δ and $\{0\}$: there exists $f \neq 0$ such that

$$f^T 0 = 0 \geq \sup_{z \in \Delta} f^T z \ \& \ f^T 0 > \inf_{z \in \Delta} f^T z.$$

In other words,

$$0 \geq \sup_{x \in S, y \in T} [f^T x - f^T y] \ \& \ 0 > \inf_{x \in S, y \in T} [f^T x - f^T y],$$

which clearly means that f separates S and T. ■

(i), Sufficiency, Step 4: Separation of nonempty convex sets with non-intersecting relative interiors. Now we are able to complete the proof of the "if" part of the Separation Theorem. Let S and T be two nonempty convex sets with non-intersecting relative interiors; we should prove that S and T can be properly separated. This is immediate: as we know from Theorem 1.1.1, the sets $S' = \operatorname{rint} S$ and $T' = \operatorname{rint} T$ are nonempty and convex; since we are given that they do not intersect, they can be separated by Step 3: there exists f such that

$$\inf_{x\in T'} f^T x \geq \sup_{y\in S'} f^T x \;\&\; \sup_{x\in T'} f^T x > \inf_{y\in S'} f^T x. \tag{1.2.13}$$

It is immediately seen that in fact f separates S and T. Indeed, the quantities in the left and the right hand sides of the first inequality in (1.2.13) clearly remain unchanged when we replace S' with $\operatorname{cl} S'$ and T' with $\operatorname{cl} T'$; by Theorem 1.1.1, $\operatorname{cl} S' = \operatorname{cl} S \supset S$ and $\operatorname{cl} T' = \operatorname{cl} T \supset T$, and we get $\inf\limits_{x\in T} f^T x = \inf\limits_{x\in T'} f^T x$, and similarly $\sup\limits_{y\in S} f^T y = \sup\limits_{y\in S'} f^T y$. Thus, we get from (1.2.13)

$$\inf_{x\in T} f^T x \geq \sup_{y\in S} f^T y.$$

It remains to note that $T' \subset T$, $S' \subset S$, so that the second inequality in (1.2.13) implies that

$$\sup_{x\in T} f^T x > \inf_{y\in S} f^T x. \quad \square$$

(ii), Necessity: prove yourself.

(ii), Sufficiency: Assuming that $\rho \equiv \inf\{\| x-y \|_2 : x \in S, y \in T\} > 0$, consider the sets $S' = \{x : \inf\limits_{y\in S} \| x-y \|_2 \leq \rho\}$. Note that S' is convex along with S (Example 1.1.4) and that $S' \cap T = \emptyset$ (why?) By (i), S' and T can be separated, and if f is a linear form which separates S' and T, then the same form strongly separates S and T (why?). The "in particular" part of (ii) readily follows from the just proved statement due to the fact that if two closed nonempty sets in $\mathbf{R}^n$ do not intersect and one of them is compact, then the sets are at positive distance from each other (why?). ∎

Exercise 1.13 *Derive the statement in Remark 1.1.1 from the Separation Theorem.*

Exercise 1.14 *Implement the following alternative approach to the proof of Separation Theorem:*

1. *Prove that if x is a point in $\mathbf{R}^n$ and S is a nonempty closed convex set in $\mathbf{R}^n$, then the problem*

$$\min_y \{\| x-y \|_2 : y \in S\}$$

   has a unique optimal solution $\bar{x}$.

2. *In the situation of 1), prove that if $x \notin S$, then the linear form $e = x - \bar{x}$ strongly separates $\{x\}$ and S:*

$$\max_{y\in S} e^T y = e^T \bar{x} = e^T x - e^T e < e^T x,$$

   thus getting a direct proof of the possibility to separate strongly a nonempty closed convex set and a point outside this set.

3. *Derive from 2) the Separation Theorem.*

1.2.6.3 Supporting hyperplanes

By the Separation Theorem, a closed and nonempty convex set M is the intersection of all closed half-spaces containing M. Among these half-spaces, the most interesting are the "extreme" ones – those with the boundary hyperplane touching M. The notion makes sense for an arbitrary (not necessary closed) convex set, but we shall use it for closed sets only, and include the requirement of closedness in the definition:

Definition 1.2.3 [Supporting plane] *Let M be a convex closed set in $\mathbf{R}^n$, and let x be a point from the relative boundary of M. A hyperplane*

$$\Pi = \{y : a^T y = a^T x\} \quad [a \neq 0]$$

is called supporting to M at x, if it separates M and $\{x\}$, i.e., if

$$a^T x \geq \sup_{y \in M} a^T y \quad \& \quad a^T x > \inf_{y \in M} a^T y. \tag{1.2.14}$$

Note that since x is a point from the relative boundary of M and therefore belongs to $\operatorname{cl} M = M$, the first inequality in (1.2.14) in fact is equality. Thus, an equivalent definition of a supporting plane is as follows:

> *Let M be a closed convex set and x be a relative boundary point of M. The hyperplane $\{y : a^T y = a^T x\}$ is called supporting to M at x, if the linear form $a(y) = a^T y$ attains its maximum on M at the point x and is nonconstant on M.*

For example, the hyperplane $\{x_1 = 1\}$ in $\mathbf{R}^n$ clearly is supporting to the unit Euclidean ball $\{x : |x| \leq 1\}$ at the point $x = e_1 = (1, 0, ..., 0)$.

The most important property of a supporting plane is its existence:

Proposition 1.2.4 [Existence of supporting hyperplane] *Let M be a convex closed set in $\mathbf{R}^n$ and x be a point from the relative boundary of M. Then*

(i) *There exists at least one hyperplane which is supporting to M at x;*

(ii) *If Π is supporting to M at x, then the intersection $M \cap \Pi$ is of affine dimension less than the one of M (recall that the affine dimension of a set is, by definition, the affine dimension of the affine hull of the set).*

Proof. (i) is easy: if x is a point from the relative boundary of M, then it is outside the relative interior of M and therefore $\{x\}$ and $\operatorname{rint} M$ can be separated by the Separation Theorem; the separating hyperplane is exactly the desired supporting to M at x hyperplane.

To prove (ii), note that if $\Pi = \{y : a^T y = a^T x\}$ is supporting to M at $x \in \partial_{\text{ri}} M$, then the set $M' = M \cap \Pi$ is a nonempty (it contains x) convex set, and the linear form $a^T y$ is constant on M' and therefore (why?) on $\operatorname{Aff}(M')$. At the same time, the form is nonconstant on M by definition of a supporting plane. Thus, $\operatorname{Aff}(M')$ is a proper (less than the entire $\operatorname{Aff}(M)$) subset of $\operatorname{Aff}(M)$, and therefore the affine dimension of $\operatorname{Aff}(M')$ (i.e., the affine dimension of M') is less than the affine dimension of $\operatorname{Aff}(M)$ (i.e., than the affine dimension of M). [2)]. ∎

1.2.7 Polar of a convex set and Dubovitski-Milutin Lemma

1.2.7.1 Polar of a convex set

Let M be a nonempty convex set in $\mathbf{R}^n$. The *polar* $\operatorname{Polar}(M)$ of M is the set of all linear forms which do not exceed 1 on M, i.e., the set of all vectors a such that $a^T x \leq 1$ for all $x \in M$:

$$\operatorname{Polar}(M) = \{a : a^T x \leq 1 \forall x \in M\}.$$

[2)] In the latter reasoning we used the following fact: if $P \subset Q$ are two affine subspaces, then the affine dimension of P is $\leq$ the one of Q, with $\leq$ being $=$ if and only if $P = Q$. Please prove this fact

For example, Polar $(\mathbf{R}^n) = \{0\}$, Polar $(\{0\}) = \mathbf{R}^n$; if L is a liner subspace in $\mathbf{R}^n$, then Polar $(L) = L^\perp$ (why?).

The following properties of the polar are evident:

1. $0 \in$ Polar (M);
2. Polar (M) is convex;
3. Polar (M) is closed.

It turns out that these properties characterize polars:

Proposition 1.2.5 *Every closed convex set M containing the origin is polar, specifically, it is polar of its polar:*

$$\begin{array}{ccc} & M \text{ is closed and convex, } 0 \in M & \\ l & \Updownarrow & \\ & M = \text{Polar}\,(\text{Polar}\,(M)) & \end{array}$$

Proof. All we need is to prove that if M is closed and convex and $0 \in M$, then $M =$ Polar $(\text{Polar}\,(M))$. By definition,

$$y \in \text{Polar}\,(M), x \in M \Rightarrow y^T x \le 1,$$

so that $M \subset$ Polar $(\text{Polar}\,(M))$. To prove that this inclusion is in fact equality, assume, on the contrary, that there exists $\bar{x} \in$ Polar $(\text{Polar}\,(M))\backslash M$. Since M is nonempty, convex and closed and $\bar{x} \notin M$, the point $\bar{x}$ can be strongly separated from M (Separation Theorem, (ii)). Thus, for appropriate b one has

$$b^T \bar{x} > \sup_{x \in M} b^T x.$$

Since $0 \in M$, the left hand side quantity in this inequality is positive; passing from b to a proportional vector $a = \lambda b$ with appropriately chosen positive λ, we may ensure that

$$a^T \bar{x} > 1 \ge \sup_{x \in M} a^T x.$$

This is the desired contradiction, since the relation $1 \ge \sup_{x \in M} a^T x$ implies that $a \in$ Polar (M), so that the relation $a^T \bar{x} > 1$ contradicts the assumption that $\bar{x} \in$ Polar $(\text{Polar}\,(M))$. ∎

Exercise 1.15 *Let M be a convex set containing the origin, and let M' be the polar of M. Prove the following facts:*

1. Polar $(M) =$ Polar $(\text{cl}\, M)$;
2. *M is bounded if and only if* $0 \in \text{int}\, M'$;
3. int $M \ne \emptyset$ *if and only if M' does not contain straight lines;*
4. *M is a closed cone of and only if M' is a closed cone. If M is a cone (not necessarily closed), then*

$$M' = \{a : a^T x \le 0 \forall x \in M\}. \tag{1.2.15}$$

1.2.7.2 Dual cone

Let $M \subset \mathbf{R}^n$ be a cone. By Exercise 1.15.4, the polar M' of M is a closed cone given by (1.2.15). The set $M_* = -M'$ (which also is a closed cone), that is, the set

$$M_* = \{a : a^T x \ge 0 \forall x \in M\}$$

of all vectors which have nonnegative inner products with all vectors from M, is called the cone *dual* to M. By Proposition 1.2.5 and Exercise 1.15.4, the family of closed cones in $\mathbf{R}^n$ is closed with respect to passing to a dual cone, and the duality is symmetric: for a closed cone M, M_* also is a closed cone, and $(M_*)_* = M$.

Exercise 1.16 *Let M be a closed cone in $\mathbf{R}^n$, and M_* be its dual cone. Prove that*

1. *M is pointed (i.e., does not contain lines) if and only M_* has a nonempty interior. Derive from this fact that M is a closed pointed cone with a nonempty interior if and only if the dual cone has the same properties.*
2. *Prove that $a \in \text{int } M_*$ if and only if $a^T x > 0$ for all nonzero vectors $x \in M$.*

1.2.7.3 Dubovitski-Milutin Lemma

Let $M_1, ..., M_k$ be cones (not necessarily closed), and M be their intersection; of course, M also is a cone. How to compute the cone dual to M?

Proposition 1.2.6 *Let $M_1, ..., M_k$ be cones. The cone M_* dual to the intersection M of the cones M_1,...,M_k contains the arithmetic sum $\widetilde{M}$ of the cones M_{1*},...,M_{k*} dual to M_1,...,M_k. If all the cones $M_1, ..., M_k$ are closed, then M_* is equal to $\text{cl}\,\widetilde{M}$. In particular, for closed cones M_1,...,M_k, M_* coincides with $\widetilde{M}$ if and only if the latter set is closed.*

Proof. Whenever $a_i \in M_{i*}$ and $x \in M$, we have $a_i^T x \geq 0$, $i = 1, ..., k$, whence $(a_1 + ... + a_k)^T x \geq 0$. Since the latter relation is valid for all $x \in M$, we conclude that $a_1 + ... + a_k \in M_*$. Thus, $\widetilde{M} \subset M_*$.

Now assume that the cones $M_1, ..., M_k$ are closed, and let us prove that $M = \text{cl}\,\widetilde{M}$. Since M_* is closed and we have seen that $\widetilde{M} \subset M_*$, all we should prove is that if $a \in M_*$, then $a \in \widehat{M} = \text{cl}\,\widetilde{M}$ as well. Assume, on the contrary, that there exists $a \in M_* \backslash \widehat{M}$. Since the set $\widetilde{M}$ clearly is a cone, its closure $\widehat{M}$ is a closed cone; by assumption, a does not belong to this closed cone and therefore, by Separation Theorem (ii), a can be strongly separated from $\widehat{M}$ and therefore – from $\widetilde{M} \subset \widehat{M}$. Thus, for some x one has

$$a^T x < \inf_{b \in \widetilde{M}} b^T x = \inf_{a_i \in M_{i*}, i=1,...,k} (a_1 + ... + a_k)^T x = \sum_{i=1}^k \inf_{a_i \in M_{i*}} a_i^T x. \tag{1.2.16}$$

From the resulting inequality it follows that $\inf\limits_{a_i \in M_{i*}} a_i^T x > -\infty$; since M_{i*} is a cone, the latter is possible if and only if $\inf\limits_{a_i \in M_{i*}} a_i^T x = 0$, i.e., if and only if for every i one has $x \in (M_{i*})_* = M_i$ (recall that the cones M_i are closed). Thus, $x \in M_i$ for all i, and the concluding quantity in (1.2.16) is 0. We see that $x \in M = \cap_i M_i$, and that (1.2.16) reduces to $a^T x < 0$. This contradicts the inclusion $a \in M_*$. ∎

Note that in general $\widetilde{M}$ can be non-closed even when all the cones $M_1, ..., M_k$ are closed. Indeed, take $k = 2$, and let M_1' be the ice-cream cone $\{(x, y, z) \in \mathbf{R}^3 : z \geq \sqrt{x^2 + y^2}\}$, and M_2' be the ray $\{z = x \leq 0, y = 0\}$ in bR^3. Observe that the points from $\widetilde{M} \equiv M_1' + M_2'$ are exactly the points of the form $(x - t, y, z - t)$ with $t \geq 0$ and $z \geq \sqrt{x^2 + y^2}$. In particular, for x positive the points $(0, 1, \sqrt{x^2 + 1} - x) = (x - x, 1, \sqrt{x^2 + 1} - x)$ belong to $\widetilde{M}$; as $x \to \infty$, these points converge to $p = (0, 1, 0)$, and thus $p \in \text{cl}\,\widetilde{M}$. On the other hand, there clearly do not exist x, y, z, t with $t \geq 0$ and $z \geq \sqrt{x^2 + y^2}$ such that $(x - t, y, z - t) = (0, 1, 0)$, that is, $p \notin \widetilde{M}$.

Dubovitski-Milutin Lemma presents a simple sufficient condition for $\widetilde{M}$ to be closed and thus to coincide with M':

Proposition 1.2.7 [Dubovitski-Milutin Lemma] *Let $M_1, ..., M_k$ be cones such that M_k is closed and the set $M_k \cap \text{int } M_1 \cap \text{int } M_2 \cap ... \cap \text{int } M_{k-1}$ is nonempty, and let $M = M_1 \cap ... \cap M_k$. Let also M_{i*} be the cones dual to M_i. Then*

(i) $\text{cl}\,M = \bigcap\limits_{i=1}^k \text{cl}\,M_i$;

(ii) *the cone $\widetilde{M} = M_{1*} + ... + M_{k*}$ is closed, and thus coincides with the cone M' dual to $\text{cl}\,M$ (or, which is the same by Exercise 1.15.1, with the cone dual to M). In other words, every linear form which is nonnegative on M can be represented as a sum of k linear forms which are nonnegative on the respective cones M_1,...,M_k.*

Proof. (i): We should prove that under the premise of the Dubovitski-Milutin Lemma, $\mathrm{cl}\, M = \bigcap_i \mathrm{cl}\, M_i$. The right hand side here contains M and is closed, so that all we should prove is that every point x in $\bigcap_{i=1}^k \mathrm{cl}\, M_i$ is the limit of an appropriate sequence $x_t \in M$. By premise of the Lemma, there exists a point $\bar{x} \in M_k \cap \mathrm{int}\, M_1 \cap \mathrm{int}\, M_2 \cap ... \cap \mathrm{int}\, M_{k-1}$; setting $x_t = t^{-1}\bar{x} + (1-t^{-1})x$, $t = 1, 2, ...$, we get a sequence converging to x as $t \to \infty$; at the same time, $x_t \in M_k$ (since x, $\bar{x}$ are in $\mathrm{cl}\, M_k = M_k$) and $x_t \in M_i$ for every $i < k$ (by Lemma 1.1.1; note that for $i < k$ one has $\bar{x} \in \mathrm{int}\, M_i$ and $x \in \mathrm{cl}\, M_i$), and thus $x_t \in M$. Thus, every point $x \in \bigcap_{i=1}^k \mathrm{cl}\, M_i$ is the limit of a sequence from M. □

(ii): To prove that under the premise of Lemma the cone $\widetilde{M} = M_{1*} + ... + M_{k*}$ is closed, let $\bar{x} \in M_k \cap \mathrm{int}\, M_1 \cap ... \mathrm{int}\, \mathcal{M}_{\|-\infty}$ (recall that we are in the case when this intersection is nonempty). and let $y_t = \sum_{i=1}^k y_{tk},$, $y_{ti} \in M_{i*}$, be a sequence of points from $\widetilde{M}$ converging, as $t \to \infty$, to certain point $\bar{y}$; we want to prove that $\bar{y} \in \widetilde{M}$. Since y_t have a limit as $t \to \infty$, the reals $p_t = y_t^T \bar{x} = \sum_{i=1}^k y_{ti}^T \bar{x}$ form a bounded sequence; since all terms in $\sum_{i=1}^k y_{ti}^T \bar{x}$ are nonnegative (due to $\bar{x} \in M_i$ and $y_{ti} \in M_{i*}$), we conclude that every one of the k sequences $\{y_{ti}^T \bar{x}\}_{i-1}^\infty$, $1 \le i \le k$, is bounded. Now let is make the following observation:

> (!) *Let K be a convex cone and $\bar{x}$ be an interior point of K, Then for certain constant C (depending on $\bar{x}$ and K) one has*
>
> $$\forall y \in K_* : \| y \|_2 \le C y^T \bar{x}. \qquad (!)$$
>
> Indeed, K contains a $\|\cdot\|_2$-ball B of certain positive radius r centered at $\bar{x}$; consequently, when $y \in K_*$, we have
>
> $$0 \le \min_{x \in B} y^T x = y^T \bar{x} - r \| y \|_2,$$
>
> that is, (!) takes place with $C = 1/r$.

By (!) and due to $\bar{x} \in \mathrm{int}\, M_i$, $i < k$, when $i < k$ the sequences y_{ti}, $t = 1, 2, ...$, are bounded along with the sequences $\{y_{ti}^T \bar{x}\}_{t=1}^\infty$; since the sequence $\{y_t = \sum_{i=1}^k y_{ti}\}_{t=1}^\infty$ is bounded, boundedness of the sequences $\{y_{ti}\}_{t=1}^\infty$ for $i = 1, ..., k-1$ implies boundedness of the sequence for $i = k$ as well. Thus, every one if the sequences $\{y_{ti} \in M_{i*}\}_{t=1}^\infty$, $1 \le i \le k$, is bounded; passing to subsequence, we can assume that $y_{ti} \to y_i$ as $t \to \infty$, $1 \le\le k$; since $y_{ti} \in M_{i*}$, we have $y_i \in M_{i*}$ (dual of a cone always is close), and recalling that $\sum_i y_{ti}$ converges to y as $t \to \infty$, we conclude that $y = \sum_i y_i \in \widetilde{M}$, as claimed. ■

1.2.8 Extreme points and Krein-Milman Theorem

Supporting planes are useful tool to prove existence of *extreme points* of convex sets. Geometrically, an extreme point of a convex set M is a point in M which cannot be obtained as a convex combination of other points of the set; and the importance of the notion comes from the fact (which we shall prove in the mean time) that the set of all extreme points of a "good enough" convex set M is the "shortest worker's instruction for building the set" – this is the smallest set of points for which M is the convex hull.

1.2.8.1 Extreme points: definition

The exact definition of an extreme point is as follows:

Definition 1.2.4 [extreme points] *Let M be a nonempty convex set in $\mathbf{R}^n$. A point $x \in M$ is called an extreme point of M, if there is no nontrivial (of positive length) segment $[u, v] \in M$ for which x is an interior point, i.e., if the relation*

$$x = \lambda u + (1-\lambda) v$$

with $\lambda \in (0, 1)$ and $u, v \in M$ is valid if and only if

$$u = v = x.$$

For example, the extreme points of a segment are exactly its endpoints; the extreme points of a triangle are its vertices; the extreme points of a (closed) circle on the 2-dimensional plane are the points of the circumference.

An equivalent definitions of an extreme point is as follows:

Exercise 1.17 *Let M be a convex set and let $x \in M$. Prove that*

1. *x is extreme if and only if the only vector h such that $x \pm h \in M$ is the zero vector;*
2. *x is extreme if and only if the set $M \backslash \{x\}$ is convex.*

1.2.8.2 Krein-Milman Theorem

It is clear that a convex set M not necessarily possesses extreme points; as an example you may take the open unit ball in $\mathbf{R}^n$. This example is not interesting – the set in question is not closed; when replacing it with its closure, we get a set (the closed unit ball) with plenty of extreme points – these are all points of the boundary. There are, however, *closed* convex sets which do not possess extreme points – e.g., a line or an affine subspace of larger dimension. A nice fact is that the absence of extreme points in a closed convex set M always has the standard reason – the set contains a line. Thus, a closed and nonempty convex set M which does not contain lines for sure possesses extreme points. And if M is nonempty convex compact set, it possesses a quite representative set of extreme points – their convex hull is the entire M.

Theorem 1.2.10 *Let M be a closed and nonempty convex set in $\mathbf{R}^n$. Then*

(i) *The set $\mathrm{Ext}(M)$ of extreme points of M is nonempty if and only if M does not contain lines;*

(ii) *If M is bounded, then M is the convex hull of its extreme points:*

$$M = \mathrm{Conv}(\mathrm{Ext}(M)),$$

so that every point of M is a convex combination of the points of $\mathrm{Ext}(M)$.

Part (ii) of this theorem is the finite-dimensional version of the famous *Krein-Milman Theorem*.

Proof. Let us start with (i). The "only if" part is easy, due to the following simple

Lemma 1.2.3 *Let M be a closed convex set in $\mathbf{R}^n$. Assume that for some $\bar{x} \in M$ and $h \in \mathbf{R}^n$ M contains the ray*

$$\{\bar{x} + th : t \geq 0\}$$

starting at $\bar{x}$ with the direction h. Then M contains also all parallel rays starting at the points of M:

$$(\forall x \in M) : \{x + th : t \geq 0\} \subset M.$$

In particular, if M contains certain line, then it contains also all parallel lines passing through the points of M.

Comment. For a closed convex set M, the set of all directions h such that $x + th \in M$ for some x and all $t \geq 0$ (i.e., by Lemma – such that $x + th \in M$ for all $x \in M$ and all $t \geq 0$) is called *the recessive cone of M* [notation: $\mathrm{Rec}(M)$]. With Lemma 1.2.3 it is immediately seen (prove it!) that $\mathrm{Rec}(M)$ indeed is a closed cone, and that

$$M + \mathrm{Rec}(M) = M.$$

Directions from $\mathrm{Rec}(M)$ are called recessive for M.

Proof of the lemma is immediate: if $x \in M$ and $\bar{x} + th \in M$ for all $t \geq 0$, then, due to convexity, for any fixed $\tau \geq 0$ we have

$$\epsilon(\bar{x} + \frac{\tau}{\epsilon}h) + (1 - \epsilon)x \in M$$

for all $\epsilon \in (0,1)$. As $\epsilon \to +0$, the left hand side tends to $x + \tau h$, and since M is closed, $x + \tau h \in M$ for every $\tau \geq 0$. □

Exercise 1.18 *Let M be a closed nonempty convex set. Prove that $\mathrm{Rec}(M) \neq \{0\}$ if and only if M is unbounded.*

Lemma 1.2.3, of course, resolves all our problems with the "only if" part. Indeed, here we should prove that if M possesses extreme points, then M does not contain lines, or, which is the same, that if M contains lines, then it has no extreme points. But the latter statement is immediate: if M contains a line, then, by Lemma, there is a line in M passing through every given point of M, so that no point can be extreme. □

Now let us prove the "if" part of (i). Thus, from now on we assume that M does not contain lines; our goal is to prove that then M possesses extreme points. Let us start with the following

Lemma 1.2.4 *Let Q be a nonempty closed convex set, $\bar{x}$ be a relative boundary point of Q and Π be a hyperplane supporting to Q at $\bar{x}$. Then all extreme points of the nonempty closed convex set $\Pi \cap Q$ are extreme points of Q.*

Proof of the Lemma. First, the set $\Pi \cap Q$ is closed and convex (as an intersection of two sets with these properties); it is nonempty, since it contains $\bar{x}$ (Π contains $\bar{x}$ due to the definition of a supporting plane, and Q contains $\bar{x}$ due to the closedness of Q). Second, let a be the linear form associated with Π:

$$\Pi = \{y : a^T y = a^T \bar{x}\},$$

so that

$$\inf_{x \in Q} a^T x < \sup_{x \in Q} a^T x = a^T \bar{x} \tag{1.2.17}$$

(see Proposition 1.2.4). Assume that y is an extreme point of $\Pi \cap Q$; what we should do is to prove that y is an extreme point of Q, or, which is the same, to prove that

$$y = \lambda u + (1-\lambda) v$$

for some $u, v \in Q$ and $\lambda \in (0,1)$ is possible only if $y = u = v$. To this end it suffices to demonstrate that under the above assumptions $u, v \in \Pi \cap Q$ (or, which is the same, to prove that $u, v \in \Pi$, since the points are known to belong to Q); indeed, we know that y is an extreme point of $\Pi \cap Q$, so that the relation $y = \lambda u + (1-\lambda) v$ with $\lambda \in (0,1)$ *and* $u, v \in \Pi \cap Q$ does imply $y = u = v$.

To prove that $u, v \in \Pi$, note that since $y \in \Pi$ we have

$$a^T y = a^T \bar{x} \geq \max\{a^T u, a^T v\}$$

(the concluding inequality follows from (1.2.17)). On the other hand,

$$a^T y = \lambda a^T u + (1-\lambda) a^T v;$$

combining these observations and taking into account that $\lambda \in (0,1)$, we conclude that

$$a^T y = a^T u = a^T v.$$

But these equalities imply that $u, v \in \Pi$. □

Equipped with the Lemma, we can easily prove (i) by induction on the dimension of the convex set M (recall that this is nothing but the affine dimension of the affine span of M, i.e., the linear dimension of the linear subspace L such that $\mathrm{Aff}(M) = a + L$).

There is nothing to do if the dimension of M is zero, i.e., if M is a point – then, of course, $M = \mathrm{Ext}(M)$. Now assume that we already have proved the nonemptiness of $\mathrm{Ext}(T)$ for all nonempty closed and not containing lines convex sets T of certain dimension k, and let us prove that the same statement is valid for the sets of dimension $k+1$. Let M be a closed convex nonempty and not containing lines set of dimension $k+1$. Since M does not contain lines and is of positive dimension, it differs from $\mathrm{Aff}(M)$ and therefore it possesses a relative boundary point $\bar{x}$ [3)]. According to Proposition 1.2.4, there exists a hyperplane $\Pi = \{x : a^T x = a^T \bar{x}\}$ which supports M at $\bar{x}$:

$$\inf_{x \in M} a^T x < \max_{x \in M} a^T x = a^T \bar{x}.$$

By the same Proposition, the set $T = \Pi \cap M$ (which is closed, convex and nonempty) is of affine dimension less than the one of M, i.e., of the dimension $\le k$. T clearly does not contain lines (since even the larger set M does not contain lines). By the inductive hypothesis, T possesses extreme points, and by Lemma 1.2.4 all these points are extreme also for M. The inductive step is completed, and (i) is proved. □

Now let us prove (ii). Thus, let M be nonempty, convex, closed and bounded; we should prove that

$$M = \mathrm{Conv}(\mathrm{Ext}(M)).$$

What is immediately seen is that the right hand side set is contained in the left hand side one. Thus, all we need is to prove that every $x \in M$ is a convex combination of points from $\mathrm{Ext}(M)$. Here we again use induction on the affine dimension of M. The case of 0-dimensional set M (i.e., a point) is trivial. Assume that the statement in question is valid for all k-dimensional convex closed and bounded sets, and let M be a convex closed and bounded set of dimension $k+1$. Let $x \in M$; to represent x as a convex combination of points from $\mathrm{Ext}(M)$, let us pass through x an arbitrary line $\ell = \{x + \lambda h : \lambda \in \mathbf{R}\}$ $(h \neq 0)$ in the affine span $\mathrm{Aff}(M)$ of M. Moving along this line from x in each of the two possible directions, we eventually leave M (since M is bounded); as it was explained in the proof of (i), it means that there exist nonnegative λ_+ and λ_- such that the points

$$\bar{x}_+ = x + \lambda_+ h,\ \bar{x}_- = x - \lambda_- h$$

both belong to the relative boundary of M. Let us verify that $\bar{x}_\pm$ are convex combinations of the extreme points of M (this will complete the proof, since x clearly is a convex combination of the two points $\bar{x}_\pm$). Indeed, M admits supporting at $\bar{x}_+$ hyperplane Π; as it was explained in the proof of (i), the set $\Pi \cap M$ (which clearly is convex, closed and bounded) is of affine dimension less than that one of M; by the inductive hypothesis, the point $\bar{x}_+$ of this set is a convex combination of extreme points of the set, and by Lemma 1.2.4 all these extreme points are extreme points of M as well. Thus, $\bar{x}_+$ is a convex combination of extreme points of M. Similar reasoning is valid for $\bar{x}_-$. ■

1.2.8.3 Example: Extreme points of a polyhedral set.

Consider a polyhedral set

$$K = \{x \in \mathbf{R}^n : Ax \le b\},$$

[3)] Indeed, there exists $z \in \mathrm{Aff}(M) \backslash M$, so that the points

$$x_\lambda = x + \lambda(z - x)$$

(x is an arbitrary fixed point of M) do not belong to M for some $\lambda \ge 1$, while $x_0 = x$ belongs to M. The set of those $\lambda \ge 0$ for which $x_\lambda \in M$ is therefore nonempty and bounded from above; this set clearly is closed (since M is closed). Thus, there exists the largest $\lambda = \lambda^*$ for which $x_\lambda \in M$. We claim that x_{λ^*} is a relative boundary point of M. Indeed, by construction this is a point from M. If it would be a point from the relative interior of M, then all the points x_λ with close to λ^* and greater than λ^* values of λ would also belong to M, which contradicts the origin of λ^*

A being a $m \times n$ matrix and b being a vector from $\mathbf{R}^m$. What are the extreme points of K? The answer is given by the following

Theorem 1.2.11 [Extreme points of polyhedral set]
Let $x \in K$. The vector x is an extreme point of K if and only if some n linearly independent (i.e., with linearly independent vectors of coefficients) inequalities of the system $Ax \leq b$ are equalities at x.

Proof. Let a_i^T, $i = 1, ..., m$, be the rows of A.

The "only if" part: let x be an extreme point of K, and let I be the set of those indices i for which $a_i^T x = b_i$; we should prove that the set F of vectors $\{a_i : i \in I\}$ contains n linearly independent vectors, or, which is the same, that $\mathrm{Lin}(F) = \mathbf{R}^n$. Assume that it is not the case; then the orthogonal complement to F contains a nonzero vector h (since the dimension of $F^\perp$ is equal to $n - \dim \mathrm{Lin}(F)$ and is therefore positive). Consider the segment $\Delta_\epsilon = [x - \epsilon h, x + \epsilon h]$, $\epsilon > 0$ being the parameter of our construction. Since h is orthogonal to the "active" vectors a_i – those with $i \in I$, all points y of this segment satisfy the relations $a_i^T y = a_i^T x = b_i$. Now, if i is a "nonactive" index – one with $a_i^T x < b_i$ – then $a_i^T y \leq b_i$ for all $y \in \Delta_\epsilon$, provided that ϵ is small enough. Since there are finitely many nonactive indices, we can choose $\epsilon > 0$ in such a way that all $y \in \Delta_\epsilon$ will satisfy all "nonactive" inequalities $a_i^T x \leq b_i$, $i \notin I$. Since $y \in \Delta_\epsilon$ satisfies, as we have seen, also all "active" inequalities, we conclude that with the above choice of ϵ we get $\Delta_\epsilon \subset K$, which is a contradiction: $\epsilon > 0$ and $h \neq 0$, so that Δ_ϵ is a nontrivial segment with the midpoint x, and no such segment can be contained in K, since x is an extreme point of K. □

To prove the "if" part, assume that $x \in K$ is such that among the inequalities $a_i^T x \leq b_i$ which are equalities at x there are n linearly independent, say, those with indices $1, ..., n$, and let us prove that x is an extreme point of K. This is immediate: assuming that x is not an extreme point, we would get the existence of a nonzero vector h such that $x \pm h \in K$. In other words, for $i = 1, ..., n$ we would have $b_i \pm a_i^T h \equiv a_i^T (x \pm h) \leq b_i$, which is possible only if $a_i^T h = 0$, $i = 1, ..., n$. But the only vector which is orthogonal to n linearly independent vectors in $\mathbf{R}^n$ is the zero vector (why?), and we get $h = 0$, which was assumed not to be the case. ■.

Corollary 1.2.1 *The set of extreme points of a polyhedral set is finite.*

Indeed, according to the above Theorem, every extreme point of a polyhedral set $K = \{x \in \mathbf{R}^n : Ax \leq b\}$ satisfies the equality version of certain n-inequality subsystem of the original system, the matrix of the subsystem being nonsingular. Due to the latter fact, an extreme point is uniquely defined by the corresponding subsystem, so that the number of extreme points does not exceed the number C_m^n of $n \times n$ submatrices of the matrix A and is therefore finite. ■

Note that C_m^n is nothing but an upper (ant typically very conservative) bound on the number of extreme points of a polyhedral set given by m inequalities in $\mathbf{R}^n$: some $n \times n$ submatrices of A can be singular and, what is more important, the majority of the nonsingular ones normally produce "candidates" which do not satisfy the remaining inequalities.

Remark 1.2.1 The result of Theorem 1.2.11 is very important, in particular, for the theory of the Simplex method – the traditional computational tool of Linear Programming. When applied to the LP program in the standard form

$$\min_x \left\{c^T x : Px = p,\ x \geq 0\right\} \quad [x \in \mathbf{R}^n],$$

with $k \times n$ matrix P, the result of Theorem 1.2.11 is that extreme points of the feasible set are exactly *the basic feasible solutions* of the system $Px = p$, i.e., nonnegative vectors x such that $Px = p$ and the set of columns of P associated with positive entries of x is linearly independent. Since the feasible set of an LP program in the standard form clearly does not contain lines, among the optimal solutions (if they exist) to an LP program in the standard form at least one is an extreme point of the feasible set (Theorem 1.2.14.(ii)). Thus, in principle we could look through the finite set of all extreme points of the feasible set (≡ through all basic feasible solutions) and to choose the one with the best value of the objective.

This recipe allows to find a feasible solution in finitely many arithmetic operations, provided that the program is solvable, and is, basically, what the Simplex method does; this latter method, of course, looks through the basic feasible solutions in a smart way which normally allows to deal with a negligible part of them only.

Another useful consequence of Theorem 1.2.11 is that if all the data in an LP program are rational, then every extreme point of the feasible domain of the program is a vector with rational entries. In particular, a solvable standard form LP program with rational data has at least one rational optimal solution.

1.2.8.3.1. Illustration: Birkhoff Theorem An $n \times n$ matrix X is called *double stochastic*, if its entries are nonnegative and all column and row sums are equal to 1. These matrices (treated as elements of $\mathbf{R}^{n^2} = \mathbf{R}^{n\times n}$) form a bounded polyhedral set, specifically, the set

$$\Pi_n = \{X = [x_{ij}]_{i,j=1}^n : x_{ij} \geq 0 \, \forall i,j, \sum_i x_{ij} = 1 \, \forall j, \sum_j x_{ij} = 1 \, \forall i\}$$

By Krein-Milman Theorem, Π_n is the convex hull of its extreme points. What are these extreme points? The answer is given by important

Theorem 1.2.12 [Birkhoff Theorem] *Extreme points of Π_n are exactly the permutation matrices of order n, i.e., $n \times n$ Boolean (i.e., with 0/1 entries) matrices with exactly one nonzero element (equal to 1) in every row and every column.*

Exercise 1.19 [Easy part] *Prove the easy part of the Theorem, specifically, that every $n \times n$ permutation matrix is an extreme point of Π_n.*

Proof of difficult part. Now let us prove that every extreme point of Π_n is a permutation matrix. To this end let us note that the $2n$ linear equations in the definition of Π_n — those saying that all row and column sums are equal to 1 - are linearly dependent, and dropping one of them, say, $\sum_i x_{in} = 1$, we do not alter the set. Indeed, the remaining equalities say that all row sums are equal to 1, so that the total sum of all entries in X is n, and that the first $n-1$ column sums are equal to 1, meaning that the last column sum is $n-(n-1) = 1$. Thus, we lose nothing when assuming that there are just $2n-1$ equality constraints in the description of Π_n. Now let us prove the claim by induction in n. The base $n = 1$ is trivial. Let us justify the inductive step $n-1 \Rightarrow n$. Thus, let X be an extreme point of Π_n. By Theorem 1.2.11, among the constraints defining Π_n (i.e., $2n-1$ equalities and n^2 inequalities $x_{ij} \geq 0$) there should be n^2 linearly independent which are satisfied at X as equations. Thus, at least $n^2 - (2n-1) = (n-1)^2$ entries in X should be zeros. It follows that at least one of the columns of X contains ≤ 1 nonzero entries (since otherwise the number of zero entries in X would be at most $n(n-2) < (n-1)^2$). Thus, there exists at least one column with at most 1 nonzero entry; since the sum of entries in this column is 1, this nonzero entry, let it be $x_{\bar{i}\bar{j}}$, is equal to 1. Since the entries in row $\bar{i}$ are nonnegative, sum up to 1 and $x_{\bar{i}\bar{j}} = 1$, $x_{\bar{i}\bar{j}} = 1$ is the only nonzero entry in its row and its column. Eliminating from X the row $\bar{i}$ and the column $\bar{j}$, we get an $(n-1)\times(n-1)$ double stochastic matrix. By inductive hypothesis, this matrix is a convex combination of $(n-1)\times(n-1)$ permutation matrices. Augmenting every one of these matrices by the column and the row we have eliminated, we get a representation of X as a convex combination of $n \times n$ permutation matrices: $X = \sum_\ell \lambda_\ell P_\ell$ with nonnegative λ_ℓ summing up to 1. Since $P_\ell \in \Pi_n$ and X is an extreme point of Π_n, in this representation all terms with nonzero coefficients λ_ℓ must be equal to $\lambda_\ell X$, so that X is one of the permutation matrices P_ℓ and as such is a permutation matrix. ∎

1.2.9 Structure of polyhedral sets

1.2.9.1 Main result

By definition, a polyhedral set M is the set of all solutions to a finite system of nonstrict linear inequalities:

$$M = \{x \in \mathbf{R}^n : Ax \leq b\}, \tag{1.2.18}$$

where A is a matrix of the column size n and certain row size m and b is m-dimensional vector. This is an "outer" description of a polyhedral set. We are about to establish an important result on the equivalent "inner" representation of a polyhedral set.

Consider the following construction. Let us take two finite nonempty set of vectors V ("vertices") and R ("rays") and build the set

$$M(V,R) = \mathrm{Conv}(V) + \mathrm{Cone}\,(R) = \{\sum_{v\in V} \lambda_v v + \sum_{r\in R} \mu_r r : \lambda_v \geq 0, \mu_r \geq 0, \sum_v \lambda_v = 1\}.$$

Thus, we take all vectors which can be represented as sums of convex combinations of the points from V and conic combinations of the points from R. The set $M(V,R)$ clearly is convex (as the arithmetic sum of two convex sets $\mathrm{Conv}(V)$ and $\mathrm{Cone}\,(R)$). The promised inner description polyhedral sets is as follows:

Theorem 1.2.13 [Inner description of a polyhedral set] *The sets of the form $M(V,R)$ are exactly the nonempty polyhedral sets: $M(V,R)$ is polyhedral, and every nonempty polyhedral set M is $M(V,R)$ for properly chosen V and R.*

The polytopes $M(V,\{0\}) = \mathrm{Conv}(V)$ are exactly the nonempty and bounded polyhedral sets. The sets of the type $M(\{0\},R)$ are exactly the polyhedral cones (sets given by finitely many nonstrict homogeneous linear inequalities).

Remark 1.2.2 In addition to the results of the Theorem, it can be proved that in the representation of a nonempty polyhedral set M as $M = \mathrm{Conv}(V) + \mathrm{Cone}\,(R)$

– the "conic" part $\mathrm{Conv}(R)$ (not the set R itself!) is uniquely defined by M and is the recessive cone of M (see Comment to Lemma 1.2.3);

– if M does not contain lines, then V can be chosen as the set of all extreme points of M.

Postponing temporarily the proof of Theorem 1.2.13, let us explain why this theorem is that important – why it is so nice to know both inner and outer descriptions of a polyhedral set.

Consider a number of natural questions:

- A. Is it true that the inverse image of a polyhedral set $M \subset \mathbf{R}^n$ under an affine mapping $y \mapsto \mathcal{P}(y) = Py + p : \mathbf{R}^m \to \mathbf{R}^n$, i.e., the set
$$\mathcal{P}^{-1}(M) = \{y \in \mathbf{R}^m : Py + p \in M\}$$
is polyhedral?
- B. Is it true that the image of a polyhedral set $M \subset \mathbf{R}^n$ under an affine mapping $x \mapsto y = \mathcal{P}(x) = Px + p : \mathbf{R}^n \to \mathbf{R}^m$ – the set
$$\mathcal{P}(M) = \{Px + p : x \in M\}$$
is polyhedral?
- C. Is it true that the intersection of two polyhedral sets is again a polyhedral set?
- D. Is it true that the arithmetic sum of two polyhedral sets is again a polyhedral set?

The answers to all these questions are positive; one way to see it is to use calculus of polyhedral representations along with the fact that polyhedrally representable sets are exactly the same as polyhedral sets (section 1.2.4). Another very instructive way is to use the just outlined results on the structure of polyhedral sets, which we intend to do now.

It is very easy to answer affirmatively to A, starting from the original – outer – definition of a polyhedral set: if $M = \{x : Ax \leq b\}$, then, of course,

$$\mathcal{P}^{-1}(M) = \{y : A(Py + p) \leq b\} = \{y : (AP)y \leq b - Ap\}$$

and therefore $\mathcal{P}^{-1}(M)$ is a polyhedral set.

An attempt to answer affirmatively to B via the same definition fails – there is no easy way to convert the linear inequalities defining a polyhedral set into those defining its image, and it is absolutely unclear why the image indeed is given by finitely many linear inequalities. Note, however, that there is no difficulty to answer affirmatively to B with the inner description of a nonempty polyhedral set: if $M = M(V, R)$, then, evidently,

$$\mathcal{P}(M) = M(\mathcal{P}(V), PR),$$

where $PR = \{Pr : r \in R\}$ is the image of R under the action of the homogeneous part of $\mathcal{P}$.

Similarly, positive answer to C becomes evident, when we use the outer description of a polyhedral set: taking intersection of the solution sets to two systems of nonstrict linear inequalities, we, of course, again get the solution set to a system of this type – you simply should put together all inequalities from the original two systems. And it is very unclear how to answer positively to D with the outer definition of a polyhedral set – what happens with inequalities when we add the solution sets? In contrast to this, the inner description gives the answer immediately:

$$\begin{array}{rcl} M(V,R) + M(V',R') & = & \mathrm{Conv}(V) + \mathrm{Cone}\,(R) + \mathrm{Conv}(V') + \mathrm{Cone}\,(R') \\ & = & [\mathrm{Conv}(V) + \mathrm{Conv}(V')] + [\mathrm{Cone}\,(R) + \mathrm{Cone}\,(R')] \\ & = & \mathrm{Conv}(V+V') + \mathrm{Cone}\,(R \cup R') \\ & = & M(V+V', R \cup R'). \end{array}$$

Note that in this computation we used two rules which should be justified: $\mathrm{Conv}(V) + \mathrm{Conv}(V') = \mathrm{Conv}(V + V')$ and $\mathrm{Cone}\,(R) + \mathrm{Cone}\,(R') = \mathrm{Cone}\,(R \cup R')$. The second is evident from the definition of the conic hull, and only the first needs simple reasoning. To prove it, note that $\mathrm{Conv}(V) + \mathrm{Conv}(V')$ is a convex set which contains $V + V'$ and therefore contains $\mathrm{Conv}(V + V')$. The inverse inclusion is proved as follows: if

$$x = \sum_i \lambda_i v_i, \;\; y = \sum_j \lambda'_j v'_j$$

are convex combinations of points from V, resp., V', then, as it is immediately seen (please check!),

$$x + y = \sum_{i,j} \lambda_i \lambda'_j (v_i + v'_j)$$

and the right hand side is a convex combination of points from $V + V'$.

We see that it is extremely useful to keep in mind both descriptions of polyhedral sets – what is difficult to see with one of them, is absolutely clear with another.

As a seemingly "more important" application of the developed theory, let us look at Linear Programming.

1.2.9.2 Theory of Linear Programming

A general Linear Programming program is the problem of maximizing a linear objective function over a polyhedral set:

$$\text{(P)} \quad \max_x \left\{ c^T x : x \in M = \{x \in \mathbf{R}^n : Ax \le b\} \right\};$$

here c is a given n-dimensional vector – the objective, A is a given $m \times n$ *constraint matrix* and $b \in \mathbf{R}^m$ is the right hand side vector. Note that (P) is called "Linear Programming program in the canonical form"; there are other equivalent forms of the problem.

1.2.9.2.1. Solvability of a Linear Programming program. According to the Linear Programming terminology which you for sure know, (P) is called

- feasible, if it admits a feasible solution, i.e., the system $Ax \le b$ is solvable, and infeasible otherwise;
- bounded, if it is feasible and the objective is above bounded on the feasible set, and unbounded, if it is feasible, but the objective is not bounded from above on the feasible set;

- solvable, if it is feasible and the optimal solution exists – the objective attains its maximum on the feasible set.

If the program is bounded, then the upper bound of the values of the objective on the feasible set is a real; this real is called the optimal value of the program and is denoted by c^*. It is convenient to assign optimal value to unbounded and infeasible programs as well – for an unbounded program it, by definition, is $+\infty$, and for an infeasible one it is $-\infty$.

Note that our terminology is aimed to deal with maximization programs; if the program is to minimize the objective, the terminology is updated in the natural way: when defining bounded/unbounded programs, we should speak about below boundedness rather than about the above boundedness of the objective, etc. For example, the optimal value of an unbounded minimization program is $-\infty$, and of an infeasible one it is $+\infty$. This terminology is consistent with the usual way of converting a minimization problem into an equivalent maximization one by replacing the original objective c with $-c$: the properties of feasibility – boundedness – solvability remain unchanged, and the optimal value in all cases changes its sign.

We have said that you for sure know the above terminology; this is not exactly true, since you definitely have heard and used the words "infeasible LP program", "unbounded LP program", but hardly used the words "bounded LP program" – only the "solvable" one. This indeed is true, although absolutely unclear in advance – a bounded LP program always is solvable. We have already established this fact, even twice — via Fourier-Motzkin elimination (section 1.2.4 and via the LP Duality Theorem). Let us reestablish this fundamental for Linear Programming fact with the tools we have at our disposal now.

Theorem 1.2.14 (i) *A Linear Programming program is solvable if and only if it is bounded.*

(ii) *If the program is solvable and the feasible set of the program does not contain lines, then at least one of the optimal solutions is an extreme point of the feasible set.*

Proof. (i): The "only if" part of the statement is tautological: the definition of solvability includes boundedness. What we should prove is the "if" part – that a bounded program is solvable. This is immediately given by the inner description of the feasible set M of the program: this is a polyhedral set, so that being nonempty (as it is for a bounded program), it can be represented as

$$M(V,R) = \mathrm{Conv}(V) + \mathrm{Cone}\,(R)$$

for some nonempty finite sets V and R. We claim first of all that since (P) is bounded, the inner product of c with every vector from R is nonpositive. Indeed, otherwise there would be $r \in R$ with $c^T r > 0$; since $M(V,R)$ clearly contains with every its point x the entire ray $\{x + tr : t \geq 0\}$, and the objective evidently is unbounded on this ray, it would be above unbounded on M, which is not the case.

Now let us choose in the *finite and nonempty* set V the point, let it be called v^*, which maximizes the objective on V. We claim that v^* is an optimal solution to (P), so that (P) is solvable. The justification of the claim is immediate: v^* clearly belongs to M; now, a generic point of $M = M(V,R)$ is

$$x = \sum_{v \in V} \lambda_v v + \sum_{r \in R} \mu_r r$$

with nonnegative λ_v and μ_r and with $\sum\limits_v \lambda_v = 1$, so that

$$\begin{array}{rcll} c^T x & = & \sum\limits_v \lambda_v c^T v + \sum\limits_r \mu_r c^T r & \\ & \leq & \sum\limits_v \lambda_v c^T v & [\text{since } \mu_r \geq 0 \text{ and } c^T r \leq 0,\ r \in R] \\ & \leq & \sum\limits_v \lambda_v c^T v^* & [\text{since } \lambda_v \geq 0 \text{ and } c^T v \leq c^T v^*] \\ & = & c^T v^* & [\text{since } \sum\limits_v \lambda_v = 1] \end{array}$$ □

(ii): if the feasible set of (P), let it be called M, does not contain lines, it, being convex and closed (as a polyhedral set) possesses extreme points. It follows that (ii) is valid in the trivial case when the

objective of (ii) is constant on the entire feasible set, since then every extreme point of M can be taken as the desired optimal solution. The case when the objective is nonconstant on M can be immediately reduced to the aforementioned trivial case: if x^* is an optimal solution to (P) and the linear form $c^T x$ is nonconstant on M, then the hyperplane $\Pi = \{x : c^T x = c^*\}$ is supporting to M at x^*; the set $\Pi \cap M$ is closed, convex, nonempty and does not contain lines, therefore it possesses an extreme point x^{**} which, on one hand, clearly is an optimal solution to (P), and on another hand is an extreme point of M by Lemma 1.2.4. ■

1.2.9.3 Structure of a polyhedral set: proofs

1.2.9.3.1. Structure of a bounded polyhedral set. Let us start with proving a significant part of Theorem 1.2.13 – the one describing *bounded* polyhedral sets.

Theorem 1.2.15 [Structure of a bounded polyhedral set] *A bounded and nonempty polyhedral set M in $\mathbf{R}^n$ is a polytope, i.e., is the convex hull of a finite nonempty set:*

$$M = M(V, \{0\}) = \mathrm{Conv}(V);$$

one can choose as V the set of all extreme points of M.

Vice versa – a polytope is a bounded and nonempty polyhedral set.

Proof. The first part of the statement – that a bounded nonempty polyhedral set is a polytope – is readily given by the Krein-Milman Theorem combined with Corollary 1.2.1. Indeed, a polyhedral set always is closed (as a set given by nonstrict inequalities involving continuous functions) and convex; if it is also bounded and nonempty, it, by the Krein-Milman Theorem, is the convex hull of the set V of its extreme points; V is finite by Corollary 1.2.1. □

Now let us prove the more difficult part of the statement – that a polytope is a bounded polyhedral set. The fact that a convex hull of a finite set is bounded is evident. Thus, all we need is to prove that the convex hull of finitely many points is a polyhedral set. To this end note that this convex hull clearly is polyhedrally representable:

$$\mathrm{Conv}\{v_1, ..., v_N\} = \{x : \exists \lambda : \lambda \geq 0, \sum_i \lambda_i = 1, x = \sum_i \lambda_i v_i\}$$

and therefore is polyhedral by Theorem 1.2.5. ■

1.2.9.3.2. Structure of a general polyhedral set: completing the proof. Now let us prove the general Theorem 1.2.13. The proof basically follows the lines of the one of Theorem 1.2.15, but with one elaboration: now we cannot use the Krein-Milman Theorem to take upon itself part of our difficulties.

To simplify language let us call VR-sets ("V" from "vertex", "R" from rays) the sets of the form $M(V, R)$, and P-sets the nonempty polyhedral sets. We should prove that every P-set is a VR-set, and vice versa. We start with proving that every P-set is a VR-set.

1.2.9.3.2.A. P⇒VR:

P⇒VR, Step 1: reduction to the case when the P-set does not contain lines. Let M be a P-set, so that M is the set of all solutions to a solvable system of linear inequalities:

$$M = \{x \in \mathbf{R}^n : Ax \leq b\} \tag{1.2.19}$$

with $m \times n$ matrix A. Such a set may contain lines; if h is the direction of a line in M, then $A(x + th) \leq b$ for some x and all $t \in \mathbf{R}$, which is possible only if $Ah = 0$. Vice versa, if h is from the kernel of A, i.e., if $Ah = 0$, then the line $x + \mathbf{R}^h$ with $x \in M$ clearly is contained in M. Thus, we come to the following fact:

Lemma 1.2.5 *ss Nonempty polyhedral set (1.2.19) contains lines if and only if the kernel of A is nontrivial, and the nonzero vectors from the kernel are exactly the directions of lines contained in M: if M contains a line with direction h, then $h \in \mathrm{Ker}\, A$, and vice versa: if $0 \neq h \in \mathrm{Ker}\, A$ and $x \in M$, then M contains the entire line $x + \mathbf{R}h$.*

Given a nonempty set (1.2.19), let us denote by L the kernel of A and by $L^\perp$ the orthogonal complement to the kernel, and let M' be the cross-section of M by $L^\perp$:

$$M' = \{x \in L^\perp : Ax \leq b\}.$$

The set M' clearly does not contain lines (since the direction of every line in M', on one hand, should belong to $L^\perp$ due to $M' \subset L^\perp$, and on the other hand – should belong to $L = \mathrm{Ker}\, A$, since a line in $M' \subset M$ is a line in M as well). The set M' is nonempty and, moreover, $M = M' + L$. Indeed, M' contains the orthogonal projections of all points from M onto $L^\perp$ (since to project a point onto $L^\perp$, you should move from this point along certain line with the direction in L, and all these movements, started in M, keep you in M by the Lemma) and therefore is nonempty, first, and is such that $M' + L \supset M$, second. On the other hand, $M' \subset M$ and $M + L = M$ by Lemma 1.2.5, whence $M' + L \subset M$. Thus, $M' + L = M$.

Finally, M' is a polyhedral set together with M, since the inclusion $x \in L^\perp$ can be represented by $\dim L$ linear equations (i.e., by $2\dim L$ nonstrict linear inequalities): you should say that x is orthogonal to $\dim L$ somehow chosen vectors $a_1, ..., a_{\dim L}$ forming a basis in L.

The results of our effort are as follows: given an arbitrary P-set M, we have represented is as the sum of a P-set M' not containing lines and a linear subspace L. With this decomposition in mind we see that in order to achieve our current goal – to prove that every P-set is a VR-set – it suffices to prove the same statement for P-sets not containing lines. Indeed, given that $M' = M(V, R')$ and denoting by R' a finite set such that $L = \mathrm{Cone}\,(R')$ (to get R', take the set of $2\dim L$ vectors $\pm a_i$, $i = 1, ..., \dim L$, where $a_1, ..., a_{\dim L}$ is a basis in L), we would obtain

$$\begin{array}{rcl} M & = & M' + L \\ & = & [\mathrm{Conv}(V) + \mathrm{Cone}\,(R)] + \mathrm{Cone}\,(R') \\ & = & \mathrm{Conv}(V) + [\mathrm{Cone}\,(R) + \mathrm{Cone}\,(R')] \\ & = & \mathrm{Conv}(V) + \mathrm{Cone}\,(R \cup R') \\ & = & M(V, R \cup R') \end{array}$$

We see that in order to establish that a P-set is a VR-set it suffices to prove the same statement for the case when the P-set in question does not contain lines.

P⇒VR, Step 2: the P-set does not contain lines. Our situation is as follows: we are given a not containing lines P-set in $\mathbf{R}^n$ and should prove that it is a VR-set. We shall prove this statement by induction on the dimension n of the space. The case of $n = 0$ is trivial. Now assume that the statement in question is valid for $n \leq k$, and let us prove that it is valid also for $n = k + 1$. Let M be a not containing lines P-set in $\mathbf{R}^{k+1}$:

$$M = \{x \in \mathbf{R}^{k+1} : a_i^T x \leq b_i,\ i = 1, ..., m\}. \tag{1.2.20}$$

Without loss of generality we may assume that all a_i are nonzero vectors (since M is nonempty, the inequalities with $a_i = 0$ are valid on the entire $\mathbf{R}^n$, and removing them from the system, we do not vary its solution set). Note that $m > 0$ – otherwise M would contain lines, since $k \geq 0$.

1^0. We may assume that M is unbounded – otherwise the desired result is given already by Theorem 1.2.15. By Exercise 1.18, there exists a recessive direction $r \neq 0$ of M Thus, M contains the ray $\{x + tr : t \geq 0\}$, whence, by Lemma 1.2.3, $M + \mathrm{Cone}\,(\{r\}) = M$. □

2^0. For every $i \leq m$, where m is the row size of the matrix A from (1.2.20), that is, the number of linear inequalities in the description of M, let us denote by M_i the corresponding "facet" of M – the

polyhedral set given by the system of inequalities (1.2.20) with the inequality $a_i^T x \le b_i$ replaced by the equality $a_i^T x = b_i$. Some of these "facets" can be empty; let I be the set of indices i of nonempty M_i's.

When $i \in I$, the set M_i is a nonempty polyhedral set – i.e., a P-set – which does not contain lines (since $M_i \subset M$ and M does not contain lines). Besides this, M_i belongs to the hyperplane $\{a_i^T x = b_i\}$, i.e., actually it is a P-set in $\mathbf{R}^k$. By the inductive hypothesis, we have representations

$$M_i = M(V_i, R_i),\ i \in I,$$

for properly chosen finite nonempty sets V_i and R_i. We claim that

$$M = M(\cup_{i\in I} V_i, \cup_{i\in I} R_i \cup \{r\}), \tag{1.2.21}$$

where r is a recessive direction of M found in 1^0; after the claim will be supported, our induction will be completed.

To prove (1.2.21), note, first of all, that the right hand side of this relation is contained in the left hand side one. Indeed, since $M_i \subset M$ and $V_i \subset M_i$, we have $V_i \subset M$, whence also $V = \cup_i V_i \subset M$; since M is convex, we have

$$\mathrm{Conv}(V) \subset M. \tag{1.2.22}$$

Further, if $r' \in R_i$, then r' is a recessive direction of M_i; since $M_i \subset M$, r' is a recessive direction of M by Lemma 1.2.3. Thus, every vector from $\cup_{i\in I} R_i$ is a recessive direction for M, same as r; thus, every vector from $R = \cup_{i\in I} R_i \cup \{r\}$ is a recessive direction of M, whence, again by Lemma 1.2.3,

$$M + \mathrm{Cone}\,(R) = M.$$

Combining this relation with (1.2.22), we get $M(V, R) \subset M$, as claimed.

It remains to prove that M is contained in the right hand side of (1.2.21). Let $x \in M$, and let us move from x along the direction $(-r)$, i.e., move along the ray $\{x - tr : t \ge 0\}$. After large enough step along this ray we leave M. (Indeed, otherwise the ray with the direction $-r$ started at x would be contained in M, while the opposite ray for sure is contained in M since r is a recessive direction of M; we would conclude that M contains a line, which is not the case by assumption.) Since the ray $\{x - tr : t \ge 0\}$ eventually leaves M and M is bounded, there exists the largest t, let it be called t^*, such that $x' = x - t^* r$ still belongs to M. It is clear that at x' one of the linear inequalities defining M becomes equality – otherwise we could slightly increase the parameter t^* still staying in M. Thus, $x' \in M_i$ for some $i \in I$. Consequently,

$$x' \in \mathrm{Conv}(V_i) + \mathrm{Cone}\,(R_i),$$

whence $x = x' + t^* r \in \mathrm{Conv}(V_i) + \mathrm{Cone}\,(R_i \cup \{r\}) \subset M(V, R)$, as claimed. □

1.2.9.3.2.B. VR⇒P: We already know that every P-set is a VR-set. Now we shall prove that every VR-set is a P-set, thus completing the proof of Theorem 1.2.13. This is immediate: a VR-set is polyhedrally representable (why?) and thus is a P-set by Theorem 1.2.5. ■

Lecture 2

Convex functions

2.1 Convex functions: first acquaintance

2.1.1 Definition and Examples

Definition 2.1.1 [convex function] *A function $f : Q \to \mathbf{R}$ defined on a nonempty subset Q of $\mathbf{R}^n$ and taking real values is called convex, if*

- *the domain Q of the function is convex;*
- *for every $x, y \in Q$ and every $\lambda \in [0, 1]$ one has*

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y). \tag{2.1.1}$$

If the above inequality is strict whenever $x \neq y$ and $0 < \lambda < 1$, f is called strictly convex.

A function f such that $-f$ is convex is called *concave*; the domain Q of a concave function should be convex, and the function itself should satisfy the inequality opposite to (2.1.1):

$$f(\lambda x + (1-\lambda)y) \ge \lambda f(x) + (1-\lambda) f(y), \quad x, y \in Q, \lambda \in [0, 1].$$

The simplest example of a convex function is an *affine function*

$$f(x) = a^T x + b$$

– the sum of a linear form and a constant. This function clearly is convex on the entire space, and the "convexity inequality" for it is equality. An affine function is both convex and concave; it is easily seen that a function which is both convex and concave on the entire space is affine.

Here are several elementary examples of "nonlinear" convex functions of one variable:

- functions convex on the whole axis:
  x^{2p}, p is a positive integer;
  $\exp\{x\}$;
- functions convex on the nonnegative ray:
  x^p, $1 \le p$;
  $-x^p$, $0 \le p \le 1$;
  $x \ln x$;
- functions convex on the positive ray:
  $1/x^p$, $p > 0$;
  $-\ln x$.

To the moment it is not clear why these functions are convex; in the mean time we shall derive a simple analytic criterion for detecting convexity which immediately demonstrates that the above functions indeed are convex.

A very convenient equivalent definition of a convex function is in terms of its *epigraph*. Given a real-valued function f defined on a nonempty subset Q of $\mathbf{R}^n$, we define its epigraph as the set

$$\mathrm{Epi}(f) = \{(t,x) \in \mathbf{R}^{n+1} : x \in Q, t \geq f(x)\};$$

geometrically, to define the epigraph, you should take the *graph* of the function – the surface $\{t = f(x), x \in Q\}$ in $\mathbf{R}^{n+1}$ – and add to this surface all points which are "above" it. The equivalent, more geometrical, definition of a convex function is given by the following simple statement (prove it!):

Proposition 2.1.1 [definition of convexity in terms of the epigraph]
A function f defined on a subset of $\mathbf{R}^n$ is convex if and only if its epigraph is a nonempty convex set in $\mathbf{R}^{n+1}$.

More examples of convex functions: norms. Equipped with Proposition 2.1.1, we can extend our initial list of convex functions (several one-dimensional functions and affine ones) by more examples – *norms*. Let $\pi(x)$ be a norm on $\mathbf{R}^n$ (see Section 1.1.2.B). To the moment we know three examples of norms – the Euclidean norm $\| x \|_2 = \sqrt{x^T x}$, the 1-norm $\| x \|_1 = \sum_i |x_i|$ and the ∞-norm $\| x \|_\infty = \max_i |x_i|$. It was also claimed (although not proved) that these are three members of an infinite family of norms

$$\| x \|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad 1 \leq p \leq \infty$$

(the right hand side of the latter relation for $p = \infty$ is, by definition, $\max_i |x_i|$).

We are about to prove that every norm is convex:

Proposition 2.1.2 *Let $\pi(x)$ be a real-valued function on $\mathbf{R}^n$ which is positively homogeneous of degree 1:*

$$\pi(tx) = t\pi(x) \quad \forall x \in \mathbf{R}^n, t \geq 0.$$

π is convex if and only if it is subadditive:

$$\pi(x+y) \leq \pi(x) + \pi(y) \quad \forall x, y \in \mathbf{R}^n.$$

In particular, a norm (which by definition is positively homogeneous of degree 1 and is subadditive) is convex.

Proof is immediate: the epigraph of a positively homogeneous of degree 1 function π clearly is a conic set: $(t,x) \in \mathrm{Epi}(\pi) \Rightarrow \lambda(t,x) \in \mathrm{Epi}(\pi)$ whenever $\lambda \geq 0$. Now, by Proposition 2.1.1 π is convex if and only if $\mathrm{Epi}(\pi)$ is convex. It is clear that a conic set is convex if and only if it contains the sum of every two its elements (why ?); this latter property is satisfied for the epigraph of a real-valued function if and only if the function is subadditive (evident). ∎

2.1.2 Elementary properties of convex functions

2.1.2.1 Jensen's inequality

The following elementary observation is, I believe, one of the most useful observations in the world:

Proposition 2.1.3 [Jensen's inequality] *Let f be convex and Q be the domain of f. Then for every convex combination*

$$\sum_{i=1}^N \lambda_i x_i$$

of points from Q one has

$$f(\sum_{i=1}^N \lambda_i x_i) \le \sum_{i=1}^N \lambda_i f(x_i).$$

Proof is immediate: the points $(f(x_i), x_i)$ clearly belong to the epigraph of f; since f is convex, its epigraph is a convex set, so that the convex combination

$$\sum_{i=1}^N \lambda_i (f(x_i), x_i) = (\sum_{i=1}^N \lambda_i f(x_i), \sum_{i=1}^N \lambda_i x_i)$$

of the points also belongs to $\mathrm{Epi}(f)$. By definition of the epigraph, the latter means exactly that $\sum_{i=1}^N \lambda_i f(x_i) \ge f(\sum_{i=1}^N \lambda_i x_i)$. ∎

Note that the definition of convexity of a function f is exactly the requirement on f to satisfy the Jensen inequality for the case of $N = 2$; we see that to satisfy this inequality for $N = 2$ is the same as to satisfy it for *all* N.

2.1.2.2 Convexity of level sets of a convex function

The following simple observation is also very useful:

Proposition 2.1.4 [convexity of level sets] *Let f be a convex function with the domain Q. Then, for every real α, the set*

$$\mathrm{lev}_\alpha(f) = \{x \in Q : f(x) \le \alpha\}$$

– the level set of f – is convex.

The proof takes one line: if $x, y \in \mathrm{lev}_\alpha(f)$ and $\lambda \in [0,1]$, then $f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y) \le \lambda\alpha + (1-\lambda)\alpha = \alpha$, so that $\lambda x + (1-\lambda)y \in \mathrm{lev}_\alpha(f)$.

Note that the convexity of level sets does *not* characterize convex functions; there are nonconvex functions which share this property (e.g., every monotone function on the axis). The "proper" characterization of convex functions in terms of convex sets is given by Proposition 2.1.1 – convex functions are exactly the functions with convex epigraphs. Convexity of level sets specify a wider family of functions, the so called *quasiconvex* ones.

2.1.3 What is the value of a convex function outside its domain?

Literally, the question which entitles this subsection is senseless. Nevertheless, when speaking about *convex* functions, it is extremely convenient to think that the function outside its domain also has a value, namely, takes the value $+\infty$; with this convention, we can say that

a convex function f on $\mathbf{R}^n$ is a function taking values in the extended real axis $\mathbf{R} \cup \{+\infty\}$ such that the domain $\mathrm{Dom}\, f$ of the function – the set of those x's where $f(x)$ is finite – is nonempty, and for all $x, y \in \mathbf{R}^n$ and all $\lambda \in [0,1]$ one has

$$f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y). \tag{2.1.2}$$

If the expression in the right hand side involves infinities, it is assigned the value according to the standard and reasonable conventions on what are arithmetic operations in the "extended real axis" $\mathbf{R} \cup \{+\infty\} \cup \{-\infty\}$:

- arithmetic operations with reals are understood in their usual sense;
- the sum of $+\infty$ and a real, same as the sum of $+\infty$ and $+\infty$ is $+\infty$; similarly, the sum of a real and $-\infty$, same as the sum of $-\infty$ and $-\infty$ is $-\infty$. The sum of $+\infty$ and $-\infty$ is undefined;

- the product of a real and $+\infty$ is $+\infty$, 0 or $-\infty$, depending on whether the real is positive, zero or negative, and similarly for the product of a real and $-\infty$. The product of two "infinities" is again infinity, with the usual rule for assigning the sign to the product.

Note that it is not clear in advance that our new definition of a convex function is equivalent to the initial one: initially we included into the definition requirement for the domain to be convex, and now we omit explicit indicating this requirement. In fact, of course, the definitions are equivalent: convexity of Dom f – i.e., the set where f is finite – is an immediate consequence of the "convexity inequality" (2.1.2).

It is convenient to think of a convex function as of something which is defined everywhere, since it saves a lot of words. For example, with this convention I can write $f + g$ (f and g are convex functions on $\mathbf{R}^n$), and everybody will understand what is meant; without this convention, I am supposed to add to this expression the explanation as follows: "$f + g$ is a function with the domain being the intersection of those of f and g, and in this intersection it is defined as $(f + g)(x) = f(x) + g(x)$".

2.2 How to detect convexity

In an optimization problem

$$\min_x \{f(x) : g_j(x) \leq 0,\ j = 1, ..., m\}$$

convexity of the objective f and the constraints g_i is crucial: it turns out that problems with this property possess nice theoretical properties (e.g., the local *necessary* optimality conditions for these problems are *sufficient for global optimality*); and what is much more important, convex problems can be efficiently (both in theoretical and, to some extent, in the practical meaning of the word) solved numerically, which is not, unfortunately, the case for general nonconvex problems. This is why it is so important to know how one can detect convexity of a given function. This is the issue we are coming to.

The scheme of our investigation is typical for mathematics. Let me start with the example which you know from Analysis. How do you detect continuity of a function? Of course, there is a definition of continuity in terms of ϵ and δ, but it would be an actual disaster if each time we need to prove continuity of a function, we were supposed to write down the proof that "for every positive ϵ there exists positive δ such that ...". In fact we use another approach: we list once for ever a number of standard operations which preserve continuity, like addition, multiplication, taking superpositions, etc., and point out a number of standard examples of continuous functions – like the power function, the exponent, etc. To prove that the operations in the list preserve continuity, same as to prove that the standard functions are continuous, this takes certain effort and indeed is done in $\epsilon - \delta$ terms; but after this effort is once invested, we normally have no difficulties with proving continuity of a given function: it suffices to demonstrate that the function can be obtained, in finitely many steps, from our "raw materials" – the standard functions which are known to be continuous – by applying our machinery – the combination rules which preserve continuity. Normally this demonstration is given by a single word "evident" or even is understood by default.

This is exactly the case with convexity. Here we also should point out the list of operations which preserve convexity and a number of standard convex functions.

2.2.1 Operations preserving convexity of functions

These operations are as follows:

- [stability under taking weighted sums] if f, g are convex functions on $\mathbf{R}^n$, then their linear combination $\lambda f + \mu g$ with *nonnegative* coefficients again is convex, provided that it is finite at least at one point;

  [this is given by straightforward verification of the definition]

- [stability under affine substitutions of the argument] the superposition $f(Ax + b)$ of a convex function f on $\mathbf{R}^n$ and affine mapping $x \mapsto Ax + b$ from $\mathbf{R}^m$ into $\mathbf{R}^n$ is convex, provided that it is finite at least at one point.

[you can prove it directly by verifying the definition or by noting that the epigraph of the superposition, if nonempty, is the inverse image of the epigraph of f under an affine mapping]

- [stability under taking pointwise sup] upper bound $\sup_\alpha f_\alpha(\cdot)$ of every family of convex functions on $\mathbf{R}^n$ is convex, provided that this bound is finite at least at one point.

  [to understand it, note that the epigraph of the upper bound clearly is the intersection of epigraphs of the functions from the family; recall that the intersection of every family of convex sets is convex]

- ["Convex Monotone superposition"] Let $f(x) = (f_1(x), ..., f_k(x))$ be vector-function on $\mathbf{R}^n$ with convex components f_i, and assume that F is a convex function on $\mathbf{R}^k$ which is monotone, i.e., such that $z \le z'$ always implies that $F(z) \le F(z')$. Then the superposition

$$\phi(x) = F(f(x)) = F(f_1(x), ..., f_k(x))$$

  is convex on $\mathbf{R}^n$, provided that it is finite at least at one point.

  Remark 2.2.1 *The expression $F(f_1(x), ..., f_k(x))$ makes no evident sense at a point x where some of f_i's are $+\infty$. By definition, we assign the superposition at such a point the value $+\infty$.*

  [To justify the rule, note that if $\lambda \in (0,1)$ and $x, x' \in \operatorname{Dom}\phi$, then $z = f(x), z' = f(x')$ are vectors from $\mathbf{R}^k$ which belong to $\operatorname{Dom} F$, and due to the convexity of the components of f we have

$$f(\lambda x + (1-\lambda)x') \le \lambda z + (1-\lambda)z';$$

  in particular, the left hand side is a vector from $\mathbf{R}^k$ – it has no "infinite entries", and we may further use the monotonicity of F:

$$\phi(\lambda x + (1-\lambda)x') = F(f(\lambda x + (1-\lambda)x')) \le F(\lambda z + (1-\lambda)z')$$

  and now use the convexity of F:

$$F(\lambda z + (1-\lambda)z') \le \lambda F(z) + (1-\lambda)F(z')$$

  to get the required relation

$$\phi(\lambda x + (1-\lambda)x') \le \lambda\phi(x) + (1-\lambda)\phi(x').$$

  ]

Imagine how many extra words would be necessary here if there were no convention on the value of a convex function outside its domain!

Two more rules are as follows:

- [stability under partial minimization] if $f(x,y) : \mathbf{R}^n_x \times \mathbf{R}^m_y$ is convex (as a function of $z = (x,y)$; this is called *joint convexity*) and the function

$$g(x) = \inf_y f(x,y)$$

  is *proper*, i.e., is $> -\infty$ everywhere and is finite at least at one point, then g is convex [this can be proved as follows. We should prove that if $x, x' \in \operatorname{Dom} g$ and $x'' = \lambda x + (1-\lambda)x'$ with $\lambda \in [0,1]$, then $x'' \in \operatorname{Dom} g$ and $g(x'') \le \lambda g(x) + (1-\lambda)g(x')$. Given positive ϵ, we can find y and y' such that $(x,y) \in \operatorname{Dom} f$, $(x',y') \in \operatorname{Dom} f$ and $g(x) + \epsilon \ge f(x,y)$, $g(y') + \epsilon \ge f(x',y')$. Taking weighted sum of these two inequalities, we get

$$\lambda g(x) + (1-\lambda)g(y) + \epsilon \ge \lambda f(x,y) + (1-\lambda)f(x',y') \ge$$

[since f is convex]

$$\geq f(\lambda x + (1-\lambda)x', \lambda y + (1-\lambda)y') = f(x'', \lambda y + (1-\lambda)y')$$

(the last $\geq$ follows from the convexity of f). The concluding quantity in the chain is $\geq g(x'')$, and we get $g(x'') \leq \lambda g(x) + (1-\lambda)g(x') + \epsilon$. In particular, $x'' \in \mathrm{Dom}\, g$ (recall that g is assumed to take only the values from $\mathbf{R}$ and the value $+\infty$). Moreover, since the resulting inequality is valid for all $\epsilon > 0$, we come to $g(x'') \leq \lambda g(x) + (1-\lambda)g(x')$, as required.]

- the "conic transformation" of a convex function f on $\mathbf{R}^n$ – the function $g(y,x) = yf(x/y)$ – is convex in the half-space $y > 0$ in $\mathbf{R}^{n+1}$.

Now we know what are the basic operations preserving convexity. Let us look what are the standard functions these operations can be applied to. A number of examples was already given, but we still do not know why the functions in the examples are convex. The usual way to check convexity of a "simple" – given by a simple formula – function is based on *differential criteria of convexity*. Let us look what are these criteria.

2.2.2 Differential criteria of convexity

From the definition of convexity of a function if immediately follows that convexity is one-dimensional property: a proper (i.e., finite at least at one point) function f on $\mathbf{R}^n$ taking values in $\mathbf{R} \cup \{+\infty\}$ is convex if and only if its restriction on every line, i.e., every function of the type $g(t) = f(x + th)$ on the axis, is either convex, or is identically $+\infty$.

It follows that to detect convexity of a function, it, in principle, suffices to know how to detect convexity of functions of one variable. This latter question can be resolved by the standard Calculus tools. Namely, in the Calculus they prove the following simple

Proposition 2.2.1 [Necessary and Sufficient Convexity Condition for smooth functions on the axis] *Let (a,b) be an interval in the axis (we do not exclude the case of $a = -\infty$ and/or $b = +\infty$). Then*

(i) A differentiable everywhere on (a,b) function f is convex on (a,b) if and only if its derivative f' is monotonically nondecreasing on (a,b);

(ii) A twice differentiable everywhere on (a,b) function f is convex on (a,b) if and only if its second derivative f'' is nonnegative everywhere on (a,b).

With the Proposition, you can immediately verify that the functions listed as examples of convex functions in Section 2.1.1 indeed are convex. The only difficulty which you may meet is that some of these functions (e.g., x^p, $p \geq 1$, and $-x^p$, $0 \leq p \leq 1$, were claimed to be convex on the half-interval $[0, +\infty)$, while the Proposition speaks about convexity of functions on intervals. To overcome this difficulty, you may use the following simple

Proposition 2.2.2 *Let M be a convex set and f be a function with $\mathrm{Dom}\, f = M$. Assume that f is convex on $\mathrm{rint}\, M$ and is continuous on M, i.e.,*

$$f(x_i) \to f(x),\ i \to \infty,$$

whenever $x_i, x \in M$ and $x_i \to x$ as $i \to \infty$. Then f is convex on M.

Proof of Proposition 2.2.1:

(i), necessity. Assume that f is differentiable and convex on (a,b); we should prove that then f' is monotonically nondecreasing. Let $x < y$ be two points of (a,b), and let us prove that $f'(x) \leq f'(y)$. Indeed, let $z \in (x,y)$. We clearly have the following representation of z as a convex combination of x and y:

$$z = \frac{y-z}{y-x}x + \frac{x-z}{y-x}y,$$

whence, from convexity,

$$f(z) \le \frac{y-z}{y-x} f(x) + \frac{x-z}{y-x} f(y),$$

whence

$$\frac{f(z)-f(x)}{x-z} \le \frac{f(y)-f(z)}{y-z}.$$

Passing here to limit as $z \to x+0$, we get

$$f'(x) \le \frac{(f(y)-f(x)}{y-x},$$

and passing in the same inequality to limit as $z \to y-0$, we get

$$f'(y) \ge \frac{(f(y)-f(x)}{y-x},$$

whence $f'(x) \le f'(y)$, as claimed. ■

(i), sufficiency. We should prove that if f is differentiable on (a,b) and f' is monotonically nondecreasing on (a,b), then f is convex on (a,b). It suffices to verify that if $x < y$, $x, y \in (a,b)$, and $z = \lambda x + (1-\lambda)y$ with $0 < \lambda < 1$, then

$$f(z) \le \lambda f(x) + (1-\lambda) f(y),$$

or, which is the same (write $f(z)$ as $\lambda f(z) + (1-\lambda) f(z)$), that

$$\frac{f(z)-f(x)}{\lambda} \le \frac{f(y)-f(z)}{1-\lambda}.$$

noticing that $z - x = \lambda(y-x)$ and $y - z = (1-\lambda)(y-x)$, we see that the inequality we should prove is equivalent to

$$\frac{f(z)-f(x)}{z-x} \le \frac{f(y)-f(z)}{y-z}.$$

But in this equivalent form the inequality is evident: by the Lagrange Mean Value Theorem, its left hand side is $f'(\xi)$ with some $\xi \in (x,z)$, while the right hand one is $f'(\eta)$ with some $\eta \in (z,y)$. Since f' is nondecreasing and $\xi \le z \le \eta$, we have $f'(\xi) \le f'(\eta)$, and the left hand side in the inequality we should prove indeed is $\le$ the right hand one. □

(ii) is immediate consequence of (i), since, as we know from the very beginning of Calculus, a differentiable function – in the case in question, it is f' – is monotonically nondecreasing on an interval if and only if its derivative is nonnegative on this interval. ■

In fact, for functions of one variable there is a differential criterion of convexity which does not assume any smoothness (we shall not prove this criterion):

Proposition 2.2.3 [convexity criterion for univariate functions]

Let $g : \mathbf{R} \to \mathbf{R} \cup \{+\infty\}$ be a function. Let the domain $\Delta = \{t : g(t) < \infty\}$ of the function be a convex set which is not a singleton, i.e., let it be an interval (a,b) with possibly added one or both endpoints ($-\infty \le a < b \le \infty$). g is convex if and only if it satisfies the following 3 requirements:

1) g is continuous on (a,b);

2) g is differentiable everywhere on (a,b), excluding, possibly, a countable set of points, and the derivative $g'(t)$ is nondecreasing on its domain;

3) at each endpoint u of the interval (a,b) which belongs to Δ g is upper semicontinuous:

$$g(u) \ge \limsup_{t \in (a,b), t \to u} g(t).$$

Proof of Proposition 2.2.2: Let $x, y \in M$ and $z = \lambda x + (1-\lambda)y$, $\lambda \in [0,1]$, and let us prove that

$$f(z) \le \lambda f(x) + (1-\lambda) f(y).$$

As we know from Theorem 1.1.1.(iii), there exist sequences $x_i \in \operatorname{rint} M$ and $y_i \in \operatorname{rint} M$ converging, respectively to x and to y. Then $z_i = \lambda x_i + (1-\lambda) y_i$ converges to z as $i \to \infty$, and since f is convex on $\operatorname{rint} M$, we have

$$f(z_i) \le \lambda f(x_i) + (1-\lambda) f(y_i);$$

passing to limit and taking into account that f is continuous on M and x_i, y_i, z_i converge, as $i \to \infty$, to $x, y, z \in M$, respectively, we obtain the required inequality. ∎

From Propositions 2.2.1.(ii) and 2.2.2 we get the following convenient *necessary and sufficient* condition for convexity of a *smooth* function of n variables:

Corollary 2.2.1 [convexity criterion for smooth functions on $\mathbf{R}^n$]
Let $f : \mathbf{R}^n \to \mathbf{R} \cup \{+\infty\}$ be a function. Assume that the domain Q of f is a convex set with a nonempty interior and that f is

- *continuous on Q*

  and

- *twice differentiable on the interior of Q.*

Then f is convex if and only if its Hessian is positive semidefinite on the interior of Q:

$$h^T f''(x) h \ge 0 \ \ \forall x \in \operatorname{int} Q \ \forall h \in \mathbf{R}^n.$$

Proof. The "only if" part is evident: if f is convex and $x \in Q' = \operatorname{int} Q$, then the function of one variable

$$g(t) = f(x + th)$$

(h is an arbitrary fixed direction in $\mathbf{R}^n$) is convex in certain neighbourhood of the point $t = 0$ on the axis (recall that affine substitutions of argument preserve convexity). Since f is twice differentiable in a neighbourhood of x, g is twice differentiable in a neighbourhood of $t = 0$, so that $g''(0) = h^T f''(x) h \ge 0$ by Proposition 2.2.1. □

Now let us prove the "if" part, so that we are given that $h^T f''(x) h \ge 0$ for every $x \in \operatorname{int} Q$ and every $h \in \mathbf{R}^n$, and we should prove that f is convex.

Let us first prove that f is convex on the interior Q' of the domain Q. By Theorem 1.1.1, Q' is a convex set. Since, as it was already explained, the convexity of a function on a convex set is one-dimensional fact, all we should prove is that every one-dimensional function

$$g(t) = f(x + t(y-x)), \ \ 0 \le t \le 1$$

(x and y are from Q') is convex on the segment $0 \le t \le 1$. Since f is continuous on $Q \supset Q'$, g is continuous on the segment; and since f is twice continuously differentiable on Q', g is continuously differentiable on $(0,1)$ with the second derivative

$$g''(t) = (y-x)^T f''(x + t(y-x))(y-x) \ge 0.$$

Consequently, g is convex on $[0,1]$ (Propositions 2.2.1.(ii) and 2.2.2). Thus, f is convex on Q'. It remains to note that f, being convex on Q' and continuous on Q, is convex on Q by Proposition 2.2.2. ∎

Applying the combination rules preserving convexity to simple functions which pass the "infinitesimal' convexity tests, we can prove convexity of many complicated functions. Consider, e.g., an *exponential posynomial* – a function

$$f(x) = \sum_{i=1}^{N} c_i \exp\{a_i^T x\}$$

with positive coefficients c_i (this is why the function is called *pos*ynomial). How could we prove that the function is convex? This is immediate:

$\exp\{t\}$ is convex (since its second order derivative is positive and therefore the first derivative is monotone, as required by the infinitesimal convexity test for smooth functions of one variable);

consequently, all functions $\exp\{a_i^T x\}$ are convex (stability of convexity under affine substitutions of argument);

consequently, f is convex (stability of convexity under taking linear combinations with nonnegative coefficients).

And if we were supposed to prove that the maximum of three posynomials is convex? Ok, we could add to our three steps the fourth, which refers to stability of convexity under taking pointwise supremum.

2.3 Gradient inequality

An extremely important property of a convex function is given by the following

Proposition 2.3.1 [Gradient inequality] *Let f be a function taking finite values and the value $+\infty$, x be an interior point of the domain of f and Q be a convex set containing x. Assume that*

- *f is convex on Q*

and

- *f is differentiable at x,*

and let $\nabla f(x)$ be the gradient of the function at x. Then the following inequality holds:

$$(\forall y \in Q): \quad f(y) \geq f(x) + (y-x)^T \nabla f(x). \tag{2.3.1}$$

Geometrically: the graph

$$\{(y,t) \in \mathbf{R}^{n+1} : y \in \mathrm{Dom}\, f \cap Q,\ t = f(y)\}$$

of the function f restricted onto the set Q is above the graph

$$\{(y,t) \in \mathbf{R}^{n+1} : t = f(x) + (y-x)^T \nabla f(x)\}$$

of the linear form tangent to f at x.

Proof. Let $y \in Q$. There is nothing to prove if $y \notin \mathrm{Dom}\, f$ (since there the right hand side in the gradient inequality is $+\infty$), same as there is nothing to prove when $y = x$. Thus, we can assume that $y \neq x$ and $y \in \mathrm{Dom}\, f$. Let us set

$$y_\tau = x + \tau(y-x),\ \ 0 < \tau \leq 1,$$

so that $y_1 = y$ and y_τ is an interior point of the segment $[x,y]$ for $0 < \tau < 1$. Now let us use the following extremely simple

Lemma 2.3.1 *Let x, x', x'' be three distinct points with $x' \in [x, x'']$, and let f be convex and finite on $[x, x'']$. Then*

$$\frac{f(x') - f(x)}{\| x' - x \|_2} \leq \frac{f(x'') - f(x)}{\| x'' - x \|_2}. \tag{2.3.2}$$

Proof of the Lemma. We clearly have

$$x' = x + \lambda(x'' - x), \quad \lambda = \frac{\| x' - x \|_2}{\| x'' - x \|_2} \in (0, 1)$$

or, which is the same,

$$x' = (1 - \lambda)x + \lambda x''.$$

From the convexity inequality

$$f(x') \le (1 - \lambda)f(x) + \lambda f(x''),$$

or, which is the same,

$$f(x') - f(x) \le \lambda(f(x'') - f(x')).$$

Dividing by λ and substituting the value of λ, we come to (2.3.2). □

Applying the Lemma to the triple $x, x' = y_\tau, x'' = y$, we get

$$\frac{f(x + \tau(y - x)) - f(x)}{\tau \| y - x \|_2} \le \frac{f(y) - f(x)}{\| y - x \|_2};$$

as $\tau \to +0$, the left hand side in this inequality, by the definition of the gradient, tends to $\| y - x \|_2^{-1} (y - x)^T \nabla f(x)$, and we get

$$\| y - x \|_2^{-1} (y - x)^T \nabla f(x) \le \| y - x \|_2^{-1} (f(y) - f(x)),$$

or, which is the same,

$$(y - x)^T \nabla f(x) \le f(y) - f(x);$$

this is exactly the inequality (2.3.1). ■

It is worthy of mentioning that in the case when Q is convex set with a nonempty interior and f is continuous on Q and differentiable on int Q, f is convex on Q if and only if the Gradient inequality (2.3.1) is valid for every pair $x \in$ int Q and $y \in Q$.

Indeed, the "only if" part, i.e., the implication

$$\textit{convexity of } f \Rightarrow \textit{Gradient inequality for all } x \in \text{int } Q \textit{ and all } y \in Q$$

is given by Proposition 2.3.1. To prove the "if" part, i.e., to establish the implication inverse to the above, assume that f satisfies the Gradient inequality for all $x \in$ int Q and all $y \in Q$, and let us verify that f is convex on Q. It suffices to prove that f is convex on the interior Q' of the set Q (see Proposition 2.2.2; recall that by assumption f is continuous on Q and Q is convex). To prove that f is convex on Q', note that Q' is convex (Theorem 1.1.1) and that, due to the Gradient inequality, on Q' f is the upper bound of the family of affine (and therefore convex) functions:

$$f(y) = \sup_{x \in Q'} f_x(y), \quad f_x(y) = f(x) + (y - x)^T \nabla f(x).$$

■

2.4 Boundedness and Lipschitz continuity of a convex function

Convex functions possess nice local properties.

Theorem 2.4.1 [local boundedness and Lipschitz continuity of convex function]
Let f be a convex function and let K be a closed and bounded set contained in the relative interior of the domain Dom f *of f. Then f is Lipschitz continuous on K – there exists constant L – the Lipschitz constant of f on K – such that*

$$|f(x) - f(y)| \leq L \parallel x - y \parallel_2 \quad \forall x, y \in K. \tag{2.4.1}$$

In particular, f is bounded on K.

Remark 2.4.1 All three assumptions on K – (1) closedness, (2) boundedness, and (3) $K \subset$ rint Dom f – are essential, as it is seen from the following three examples:

- $f(x) = 1/x$, Dom $F = (0, +\infty)$, $K = (0, 1]$. We have (2), (3) but not (1); f is neither bounded, nor Lipschitz continuous on K.
- $f(x) = x^2$, Dom $f = \mathbf{R}$, $K = \mathbf{R}$. We have (1), (3) and not (2); f is neither bounded nor Lipschitz continuous on K.
- $f(x) = -\sqrt{x}$, Dom $f = [0, +\infty)$, $K = [0, 1]$. We have (1), (2) and not (3); f is not Lipschitz continuous on K [1)], although is bounded. With properly chosen convex function f of two variables and non-polyhedral compact domain (e.g., with Dom f being the unit circle), we could demonstrate also that lack of (3), even in presence of (1) and (2), may cause unboundedness of f at K as well.

Remark 2.4.2 *Theorem 2.4.1 says that a convex function f is bounded on every compact (i.e., closed and bounded) subset of the relative interior of* Dom f. *In fact there is much stronger statement on the below boundedness of f: f is below bounded on any bounded subset of $\mathbf{R}^n$!.*

Proof of Theorem 2.4.1. We shall start with the following local version of the Theorem.

Proposition 2.4.1 *Let f be a convex function, and let $\bar{x}$ be a point from the relative interior of the domain* Dom f *of f. Then*
(i) *f is bounded at $\bar{x}$: there exists a positive r such that f is bounded in the r-neighbourhood $U_r(\bar{x})$ of $\bar{x}$ in the affine span of* Dom f*:*

$$\exists r > 0, C: \quad |f(x)| \leq C \ \ \forall x \in U_r(\bar{x}) = \{x \in \mathrm{Aff}(\mathrm{Dom}\, f) : \parallel x - \bar{x} \parallel_2 \leq r\};$$

(ii) *f is Lipschitz continuous at $\bar{x}$, i.e., there exists a positive ρ and a constant L such that*

$$|f(x) - f(x')| \leq L \parallel x - x' \parallel_2 \ \ \forall x, x' \in U_\rho(\bar{x}).$$

Implication "Proposition 2.4.1 $\Rightarrow$ Theorem 2.4.1" is given by standard Analysis reasoning. All we need is to prove that if K is a bounded and closed (i.e., a compact) subset of rint Dom f, then f is Lipschitz continuous on K (the boundedness of f on K is an evident consequence of its Lipschitz continuity on K and boundedness of K). Assume, on contrary, that f is not Lipschitz continuous on K; then for every integer i there exists a pair of points $x_i, y_i \in K$ such that

$$f(x_i) - f(y_i) \geq i \parallel x_i - y_i \parallel_2 . \tag{2.4.2}$$

Since K is compact, passing to a subsequence we can ensure that $x_i \to x \in K$ and $y_i \to y \in K$. By Proposition 2.4.1 the case $x = y$ is impossible – by Proposition f is Lipschitz continuous in a neighbourhood B of $x = y$; since $x_i \to x, y_i \to y$, this neighbourhood should contain all x_i and y_i with large enough indices i; but then, from the Lipschitz continuity of f in B, the ratios $(f(x_i) - f(y_i))/ \parallel$

[1)] indeed, we have $\lim\limits_{t \to +0} \frac{f(0) - f(t)}{t} = \lim\limits_{t \to +0} t^{-1/2} = +\infty$, while for a Lipschitz continuous f the ratios $t^{-1}(f(0) - f(t))$ should be bounded

$x_i - y_i \|_2$ form a bounded sequence, which we know is not the case. Thus, the case $x = y$ is impossible. The case $x \neq y$ is "even less possible" – since, by Proposition, f is continuous on $\mathrm{Dom}\, f$ at both the points x and y (note that Lipschitz continuity at a point clearly implies the usual continuity at it), so that we would have $f(x_i) \to f(x)$ and $f(y_i) \to f(y)$ as $i \to \infty$. Thus, the left hand side in (2.4.2) remains bounded as $i \to \infty$. In the right hand side one factor – i – tends to ∞, and the other one has a nonzero limit $\| x - y \|$, so that the right hand side tends to ∞ as $i \to \infty$; this is the desired contradiction. □

Proof of Proposition 2.4.1.

1^0. We start with proving the *above boundedness* of f in a neighbourhood of $\bar{x}$. This is immediate: we know that there exists a neighbourhood $U_{\bar{r}}(\bar{x})$ which is contained in $\mathrm{Dom}\, f$ (since, by assumption, $\bar{x}$ is a relative interior point of $\mathrm{Dom}\, f$). Now, we can find a small simplex Δ of the dimension $m = \dim \mathrm{Aff}(\mathrm{Dom}\, f)$ with the vertices $x_0, ..., x_m$ in $U_{\bar{r}}(\bar{x})$ in such a way that $\bar{x}$ will be a convex combination of the vectors x_i with *positive* coefficients, even with the coefficients $1/(m+1)$:

$$\bar{x} = \sum_{i=0}^{m} \frac{1}{m+1} x_i \quad {}^{2)}.$$

We know that $\bar{x}$ is the point from the relative interior of Δ (Exercise 1.8); since Δ spans the same affine subspace as $\mathrm{Dom}\, f$, it means that Δ contains $U_r(\bar{x})$ with certain $r > 0$. Now, we have

$$\Delta = \{\sum_{i=0}^{m} \lambda_i x_i : \lambda_i \geq 0, \sum_i \lambda_i = 1\}$$

so that in Δ f is bounded from above by the quantity $\max\limits_{0 \leq i \leq m} f(x_i)$ by Jensen's inequality:

$$f(\sum_{i=0}^{m} \lambda_i x_i) \leq \sum_{i=0}^{m} \lambda_i f(x_i) \leq \max_i f(x_i).$$

Consequently, f is bounded from above, by the same quantity, in $U_r(\bar{x})$.

2^0. Now let us prove that if f is above bounded, by some C, in $U_r = U_r(\bar{x})$, then it in fact is below bounded in this neighbourhood (and, consequently, is bounded in U_r). Indeed, let $x \in U_r$, so that $x \in \mathrm{Aff}(\mathrm{Dom}\, f)$ and $\| x - \bar{x} \|_2 \leq r$. Setting $x' = \bar{x} - [x - \bar{x}] = 2\bar{x} - x$, we get $x' \in \mathrm{Aff}(\mathrm{Dom}\, f)$ and $\| x' - \bar{x} \|_2 = \| x - \bar{x} \|_2 \leq r$, so that $x' \in U_r$. Since $\bar{x} = \frac{1}{2}[x + x']$, we have

$$2f(\bar{x}) \leq f(x) + f(x'),$$

whence

$$f(x) \geq 2f(\bar{x}) - f(x') \geq 2f(\bar{x}) - C, \;\; x \in U_r(\bar{x}),$$

and f is indeed below bounded in U_r.

(i) is proved.

3^0. (ii) is an immediate consequence of (i) and Lemma 2.3.1. Indeed, let us prove that f is Lipschitz continuous in the neighbourhood $U_{r/2}(\bar{x})$, where $r > 0$ is such that f is bounded in $U_r(\bar{x})$ (we already know from (i) that the required r does exist). Let $|f| \leq C$ in U_r, and let $x, x' \in U_{r/2}$, $x \neq x'$. Let

[2] to see that the required Δ exists, let us act as follows: first, the case of $\mathrm{Dom}\, f$ being a singleton is evident, so that we can assume that $\mathrm{Dom}\, f$ is a convex set of dimension $m \geq 1$. Without loss of generality, we may assume that $\bar{x} = 0$, so that $0 \in \mathrm{Dom}\, f$ and therefore $\mathrm{Aff}(\mathrm{Dom}\, f) = \mathrm{Lin}(\mathrm{Dom}\, f)$. By Linear Algebra, we can find m vectors $y_1, ..., y_m$ in $\mathrm{Dom}\, f$ which form a basis in $\mathrm{Lin}(\mathrm{Dom}\, f) = \mathrm{Aff}(\mathrm{Dom}\, f)$. Setting $y_0 = -\sum\limits_{i=1}^{m} y_i$ and taking into account that $0 = \bar{x} \in \mathrm{rint}\, \mathrm{Dom}\, f$, we can find $\epsilon > 0$ such that the vectors $x_i = \epsilon y_i$, $i = 0, ..., m$, belong to $U_{\bar{r}}(\bar{x})$. By construction, $\bar{x} = 0 = \frac{1}{m+1} \sum\limits_{i=0}^{m} x_i$.

us extend the segment $[x, x']$ through the point x' until it reaches, at certain point x'', the (relative) boundary of U_r. We have

$$x' \in (x, x''); \quad \| x'' - \bar{x} \|_2 = r.$$

From (2.3.2) we have

$$f(x') - f(x) \leq \| x' - x \|_2 \frac{f(x'') - f(x)}{\| x'' - x \|_2}.$$

The second factor in the right hand side does not exceed the quantity $(2C)/(r/2) = 4C/r$; indeed, the numerator is, in absolute value, at most $2C$ (since $|f|$ is bounded by C in U_r and both x, x'' belong to U_r), and the denominator is at least $r/2$ (indeed, x is at the distance at most $r/2$ from $\bar{x}$, and x'' is at the distance exactly r from $\bar{x}$, so that the distance between x and x'', by the triangle inequality, is at least $r/2$). Thus, we have

$$f(x') - f(x) \leq (4C/r) \| x' - x \|_2, \ x, x' \in U_{r/2};$$

swapping x and x', we come to

$$f(x) - f(x') \leq (4C/r) \| x' - x \|_2,$$

whence

$$|f(x) - f(x')| \leq (4C/r) \| x - x' \|_2, \ x, x' \in U_{r/2},$$

as required in (ii). ■

2.5 Maxima and minima of convex functions

As it was already mentioned, optimization problems involving convex functions possess nice theoretical properties. One of the most important of these properties is given by the following

Theorem 2.5.1 ["Unimodality"] *Let f be a convex function on a convex set $Q \subset \mathbf{R}^n$, and let $x^* \in Q \cap \mathrm{Dom}\, f$ be a local minimizer of f on Q:*

$$(\exists r > 0): \quad f(y) \geq f(x^*) \quad \forall y \in Q, \ \| y - x \|_2 < r. \tag{2.5.1}$$

Then x^* is a global minimizer of f on Q:

$$f(y) \geq f(x^*) \quad \forall y \in Q. \tag{2.5.2}$$

Moreover, the set $\operatorname{Argmin}_Q f$ of all local ($\equiv$ global) minimizers of f on Q is convex.

If f is strictly convex (i.e., the convexity inequality $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$ is strict whenever $x \neq y$ and $\lambda \in (0,1)$), then the above set is either empty or is a singleton.

Proof. 1) Let x^* be a local minimizer of f on Q and $y \in Q$, $y \neq x^*$; we should prove that $f(y) \geq f(x^*)$. There is nothing to prove if $f(y) = +\infty$, so that we may assume that $y \in \mathrm{Dom}\, f$. Note that also $x^* \in \mathrm{Dom}\, f$ for sure – by definition of a local minimizer.

For all $\tau \in (0,1)$ we have, by Lemma 2.3.1,

$$\frac{f(x^* + \tau(y - x^*)) - f(x^*)}{\tau \| y - x^* \|_2} \leq \frac{f(y) - f(x^*)}{\| y - x^* \|_2}.$$

Since x^* is a local minimizer of f, the left hand side in this inequality is nonnegative for all small enough values of $\tau > 0$. We conclude that the right hand side is nonnegative, i.e., $f(y) \geq f(x^*)$. □

2) To prove convexity of $\operatorname{Argmin}_Q f$, note that $\operatorname{Argmin}_Q f$ is nothing but the level set $\mathrm{lev}_\alpha(f)$ of f associated with the minimal value $\min_Q f$ of f on Q; as a level set of a convex function, this set is convex (Proposition 2.1.4).

3) To prove that the set $\operatorname{Argmin}_Q f$ associated with a strictly convex f is, if nonempty, a singleton, note that if there were two distinct minimizers x', x'', then, from strict convexity, we would have

$$f(\frac{1}{2}x' + \frac{1}{2}x'') < \frac{1}{2}[f(x') + f(x'') == \min_Q f,$$

which clearly is impossible - the argument in the left hand side is a point from Q! ■

Another pleasant fact is that in the case of differentiable convex functions the known from Calculus necessary optimality condition (the Fermat rule) is sufficient for global optimality:

Theorem 2.5.2 [Necessary and sufficient optimality condition for a differentiable convex function]
Let f be convex function on convex set $Q \subset \mathbf{R}^n$, and let x^* be an interior point of Q. Assume that f is differentiable at x^*. Then x^* is a minimizer of f on Q if and only if

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

Proof. As a *necessary* condition for local optimality, the relation $\nabla f(x^*) = 0$ is known from Calculus; it has nothing in common with convexity. The essence of the matter is, of course, the *sufficiency* of the condition $\nabla f(x^*) = 0$ for *global optimality* of x^* in the case of *convex* f. This sufficiency is readily given by the Gradient inequality (2.3.1): by virtue of this inequality and due to $\nabla f(x^*) = 0$,

$$f(y) \geq f(x^*) + (y - x^*)\nabla f(x^*) = f(x^*)$$

for all $y \in Q$. ■

A natural question is what happens if x^* in the above statement is not necessarily an interior point of Q. Thus, assume that x^* is an arbitrary point of a convex set Q and that f is convex on Q and differentiable at x^* (the latter means exactly that $\operatorname{Dom} f$ contains a neighbourhood of x^* and f possesses the first order derivative at x^*). Under these assumptions, when x^* is a minimizer of f on Q?

The answer is as follows: let

$$T_Q(x^*) = \{h \in \mathbf{R}^n : x^* + th \in Q \quad \forall \text{ small enough } t > 0\}$$

be the *radial cone* of Q at x^*; geometrically, this is the set of all directions leading from x^* inside Q, so that a small enough positive step from x^* along the direction keeps the point in Q. From the convexity of Q it immediately follows that the radial cone indeed is a convex cone (not necessary closed). For example, when x^* is an interior point of Q, then the radial cone to Q at x^* clearly is the entire $\mathbf{R}^n$. A more interesting example is the radial cone to a polyhedral set

$$Q = \{x : a_i^T x \leq b_i,\ i = 1, ..., m\}; \tag{2.5.3}$$

for $x^* \in Q$ the corresponding radial cone clearly is the polyhedral cone

$$\{h : a_i^T h \leq 0 \quad \forall i : a_i^T x^* = b_i\} \tag{2.5.4}$$

corresponding to the *active* at x^* (i.e., satisfied at the point as equalities rather than as strict inequalities) constraints $a_i^T x \leq b_i$ from the description of Q.

Now, for the functions in question the necessary and sufficient condition for x^* to be a minimizer of f on Q is as follows:

Proposition 2.5.1 *Let Q be a convex set, let $x^* \in Q$, and let f be a convex on Q function which is differentiable at x^*. The necessary and sufficient condition for x^* to be a minimizer of f on Q is that the derivative of f taken at x^* along every direction from $T_Q(x^*)$ should be nonnegative:*

$$h^T \nabla f(x^*) \geq 0 \quad \forall h \in T_Q(x^*).$$

Proof is immediate. The necessity is an evident fact which has nothing in common with convexity: assuming that x^* is a local minimizer of f on Q, we note that if there were $h \in T_Q(x^*)$ with $h^T \nabla f(x^*) < 0$, then we would have

$$f(x^* + th) < f(x^*)$$

for all small enough positive t. On the other hand, $x^* + th \in Q$ for all small enough positive t due to $h \in T_Q(x^*)$. Combining these observations, we conclude that in every neighbourhood of x^* there are points from Q with strictly better than the one at x^* values of f; this contradicts the assumption that x^* is a local minimizer of f on Q.

The sufficiency is given by the Gradient Inequality, exactly as in the case when x^* is an interior point of Q. ∎

Proposition 2.5.1 says that whenever f is convex on Q and differentiable at $x^* \in Q$, the necessary and sufficient condition for x^* to be a minimizer of f on Q is that the linear form given by the gradient $\nabla f(x^*)$ of f at x^* should be nonnegative at all directions from the radial cone $T_Q(x^*)$. The linear forms nonnegative at all directions from the radial cone also form a cone; it is called the cone *normal* to Q at x^* and is denoted $N_Q(x^*)$. Thus, Proposition says that the necessary and sufficient condition for x^* to minimize f on Q is the inclusion $\nabla f(x^*) \in N_Q(x^*)$. What does this condition actually mean, it depends on what is the normal cone: whenever we have an explicit description of it, we have an explicit form of the optimality condition.

For example, when $T_Q(x^*) = \mathbf{R}^n$ (it is the same as to say that x^* is an interior point of Q), then the normal cone is comprised of the linear forms nonnegative at the entire space, i.e., it is the trivial cone $\{0\}$; consequently, for the case in question the optimality condition becomes the Fermat rule $\nabla f(x^*) = 0$, as we already know.

When Q is the polyhedral set (2.5.3), the normal cone is the polyhedral cone (2.5.4); it is comprised of all directions which have nonpositive inner products with all a_i coming from the active, in the aforementioned sense, constraints. The normal cone is comprised of all vectors which have nonnegative inner products with all these directions, i.e., of vectors a such that the inequality $h^T a \geq 0$ is a consequence of the inequalities $h^T a_i \leq 0$, $i \in I(x^*) \equiv \{i : a_i^T x^* = b_i\}$. From the Homogeneous Farkas Lemma we conclude that the normal cone is simply the conic hull of the vectors $-a_i$, $i \in I(x^*)$. Thus, in the case in question (*) reads:

$x^* \in Q$ is a minimizer of f on Q if and only if there exist nonnegative reals λ_i^* associated with "active" (those from $I(x^*)$) values of i such that

$$\nabla f(x^*) + \sum_{i \in I(x^*)} \lambda_i^* a_i = 0.$$

These are the famous *Karush-Kuhn-Tucker* optimality conditions; these conditions are necessary for optimality in an essentially wider situation.

The indicated results demonstrate that the fact that a point $x^* \in \operatorname{Dom} f$ is a global minimizer of a convex function f depends only on the local behaviour of f at x^*. This is not the case with maximizers of a convex function. First of all, such a maximizer, if exists, in all nontrivial cases should belong to the boundary of the domain of the function:

Theorem 2.5.3 *Let f be convex, and let Q be the domain of f. Assume that f attains its maximum on Q at a point x^* from the relative interior of Q. Then f is constant on Q.*

Proof. Let $y \in Q$; we should prove that $f(y) = f(x^*)$. There is nothing to prove if $y = x^*$, so that we may assume that $y \neq x^*$. Since, by assumption, $x^* \in \operatorname{rint} Q$, we can extend the segment $[x^*, y]$ through the endpoint x^*, keeping the left endpoint of the segment in Q; in other words, there exists a point $y' \in Q$ such that x^* is an interior point of the segment $[y', y]$:

$$x^* = \lambda y' + (1 - \lambda) y$$

for certain $\lambda \in (0,1)$. From the definition of convexity

$$f(x^*) \le \lambda f(y') + (1-\lambda) f(y).$$

Since both $f(y')$ and $f(y)$ do not exceed $f(x^*)$ (x^* is a maximizer of f on Q!) and both the weights λ and $1-\lambda$ are strictly positive, the indicated inequality can be valid only if $f(y') = f(y) = f(x^*)$. ∎

The next theorem gives further information on maxima of convex functions:

Theorem 2.5.4 *Let f be a convex function on $\mathbf{R}^n$ and E be a subset of $\mathbf{R}^n$. Then*

$$\sup_{\mathrm{Conv}\, E} f = \sup_E f. \tag{2.5.5}$$

In particular, if $S \subset \mathbf{R}^n$ is convex and compact set, then the supremum of f on S is equal to the supremum of f on the set of extreme points of S:

$$\sup_S f = \sup_{\mathrm{Ext}(S)} f \tag{2.5.6}$$

Proof. To prove (2.5.5), let $x \in \mathrm{Conv}\, E$, so that x is a convex combination of points from E (Theorem 1.1.4 on the structure of convex hull):

$$x = \sum_i \lambda_i x_i \quad [x_i \in E,\ \lambda_i \ge 0,\ \sum_i \lambda_i = 1].$$

Applying Jensen's inequality (Proposition 2.1.3), we get

$$f(x) \le \sum_i \lambda_i f(x_i) \le \sum_i \lambda_i \sup_E f = \sup_E f,$$

so that the left hand side in (2.5.5) is $\le$ the right hand one; the inverse inequality is evident, since $\mathrm{Conv}\, E \supset E$. □

To derive (2.5.6) from (2.5.5), it suffices to note that from the Krein-Milman Theorem (Theorem 1.2.10) for a convex compact set S one has $S = \mathrm{Conv}\, \mathrm{Ext}(S)$. ∎

The last theorem on maxima of convex functions is as follows:

Theorem 2.5.5 * *Let f be a convex function such that the domain Q of f is closed and does not contain lines. Then*

(i) *If the set*

$$\mathop{\mathrm{Argmax}}_Q f \equiv \{x \in Q : f(x) \ge f(y)\ \forall y \in Q\}$$

of global maximizers of f is nonempty, then it intersects the set $\mathrm{Ext}(Q)$ of the extreme points of Q, so that at least one of the maximizers of f is an extreme point of Q;

(ii) *If the set Q is polyhedral and f is above bounded on Q, then the maximum of f on Q is achieved:* $\mathop{\mathrm{Argmax}}_Q f \ne \emptyset$.

Proof. Let us start with (i). We shall prove this statement by induction on the dimension of Q. The base $\dim Q = 0$, i.e., the case of a singleton Q, is trivial, since here $Q = \mathrm{Ext}Q = \mathop{\mathrm{Argmax}}_Q f$. Now assume that the statement is valid for the case of $\dim Q \le p$, and let us prove that it is valid also for the case of $\dim Q = p+1$. Let us first verify that the set $\mathop{\mathrm{Argmax}}_Q f$ intersects with the (relative) boundary of Q. Indeed, let $x \in \mathop{\mathrm{Argmax}}_Q f$. There is nothing to prove if x itself is a relative boundary point of Q; and if x is not a boundary point, then, by Theorem 2.5.3, f is constant on Q, so that $\mathop{\mathrm{Argmax}}_Q f = Q$; and since Q is

closed, every relative boundary point of Q (such a point does exist, since Q does not contain lines and is of positive dimension) is a maximizer of f on Q, so that here again $\operatorname*{Argmax}_Q f$ intersects $\partial_{\rm ri}\, Q$.

Thus, among the maximizers of f there exists at least one, let it be x, which belongs to the relative boundary of Q. Let H be the hyperplane which supports Q at x (see Section 1.2.6), and let $Q' = Q \cap H$. The set Q' is closed and convex (since Q and H are), nonempty (it contains x) and does not contain lines (since Q does not). We have $\max\limits_Q f = f(x) = \max\limits_{Q'} f$ (note that $Q' \subset Q$), whence

$$\emptyset \neq \operatorname*{Argmax}_{Q'} f \subset \operatorname*{Argmax}_Q f.$$

Same as in the proof of the Krein-Milman Theorem (Theorem 1.2.10), we have $\dim Q' < \dim Q$. In view of this inequality we can apply to f and Q' our inductive hypothesis to get

$$\operatorname{Ext}(Q') \cap \operatorname*{Argmax}_{Q'} f \neq \emptyset.$$

Since $\operatorname{Ext}(Q') \subset \operatorname{Ext}(Q)$ by Lemma 1.2.4 and, as we just have seen, $\operatorname*{Argmax}_{Q'} f \subset \operatorname*{Argmax}_Q f$, we conclude that the set $\operatorname{Ext}(Q) \cap \operatorname*{Argmax}_Q f$ is not smaller than $\operatorname{Ext}(Q') \cap \operatorname*{Argmax}_{Q'} f$ and is therefore nonempty, as required. □

To prove (ii), let us use the known to us from Lecture 4 results on the structure of a polyhedral convex set:

$$Q = \operatorname{Conv}(V) + \operatorname{Cone}(R),$$

where V and R are finite sets. We are about to prove that the upper bound of f on Q is exactly the maximum of f on the finite set V:

$$\forall x \in Q: \quad f(x) \leq \max_{v \in V} f(v). \tag{2.5.7}$$

This will mean, in particular, that f attains its maximum on Q – e.g., at the point of V where f attains its maximum on V.

To prove the announced statement, I first claim that if f is above bounded on Q, then every direction $r \in \operatorname{Cone}(R)$ is *descent* for f, i.e., is such that every step in this direction taken from every point $x \in Q$ decreases f:

$$f(x + tr) \leq f(x) \quad \forall x \in Q \forall t \geq 0. \tag{2.5.8}$$

Indeed, if, on contrary, there were $x \in Q$, $r \in R$ and $t \geq 0$ such that $f(x + tr) > f(x)$, we would have $t > 0$ and, by Lemma 2.3.1,

$$f(x + sr) \geq f(x) + \frac{s}{t}(f(x + tr) - f(x)), \ s \geq t.$$

Since $x \in Q$ and $r \in \operatorname{Cone}(R)$, $x + sr \in Q$ for all $s \geq 0$, and since f is above bounded on Q, the left hand side in the latter inequality is above bounded, while the right hand one, due to $f(x + tr) > f(x)$, goes to $+\infty$ as $s \to \infty$, which is the desired contradiction.

Now we are done: to prove (2.5.7), note that a generic point $x \in Q$ can be represented as

$$x = \sum_{v \in V} \lambda_v v + r \quad [r \in \operatorname{Cone}(R); \sum_v \lambda_v = 1, \lambda_v \geq 0],$$

and we have

$$\begin{array}{rcll} f(x) & = & f(\sum\limits_{v\in V}\lambda_v v + r) & \\ & \le & f(\sum\limits_{v\in V}\lambda_v v) & \text{[by (2.5.8)]} \\ & \le & \sum\limits_{v\in V}\lambda_v f(v) & \text{[Jensen's Inequality]} \\ & \le & \max\limits_{v\in V} f(v) & \blacksquare \end{array}$$

2.6 Subgradients and Legendre transformation

2.6.1 Proper functions and their representation

According to one of two equivalent definitions, a convex function f on $\mathbf{R}^n$ is a function taking values in $\mathbf{R}\cup\{+\infty\}$ such that the epigraph

$$\mathrm{Epi}(f) = \{(t,x)\in\mathbf{R}^{n+1} : t \ge f(x)\}$$

is a nonempty convex set. Thus, there is no essential difference between convex functions and convex sets: convex function generates a convex set – its epigraph – which of course remembers everything about the function. And the only specific property of the epigraph as a convex set is that it has a recessive direction – namely, $e=(1,0)$ – such that the intersection of the epigraph with every line directed by h is either empty, or is a closed ray. Whenever a nonempty convex set possesses such a property with respect to certain direction, it can be represented, in properly chosen coordinates, as the epigraph of some convex function. Thus, a convex function is, basically, nothing but a way to look, in the literal meaning of the latter verb, at a convex set.

Now, we know that "actually good" convex sets are closed ones: they possess a lot of important properties (e.g., admit a good outer description) which are not shared by arbitrary convex sets. It means that among convex functions there also are "actually good" ones – those with closed epigraphs. Closedness of the epigraph can be "translated" to the functional language and there becomes a special kind of continuity – *lower semicontinuity:*

Definition 2.6.1 [Lower semicontinuity] *Let f be a function (not necessarily convex) defined on $\mathbf{R}^n$ and taking values in $\mathbf{R}\cup\{+\infty\}$. We say that f is lower semicontinuous at a point $\bar{x}$, if for every sequence of points $\{x_i\}$ converging to $\bar{x}$ one has*

$$f(\bar{x}) \le \lim\inf_{i\to\infty} f(x_i)$$

(here, of course, $\lim\inf$ *of a sequence with all terms equal to $+\infty$ is $+\infty$).*

f is called lower semicontinuous, if it is lower semicontinuous at every point.

A trivial example of a lower semicontinuous function is a continuous one. Note, however, that a semicontinuous function is not obliged to be continuous; what it is obliged, is to make only "jumps down". For example, the function

$$f(x) = \begin{cases} 0, & x\ne 0 \\ a, & x = 0 \end{cases}$$

is lower semicontinuous if $a\le 0$ ("jump down at $x=0$ or no jump at all"), and is not lower semicontinuous if $a>0$ ("jump up").

The following statement links lower semicontinuity with the geometry of the epigraph:

Proposition 2.6.1 *A function f defined on $\mathbf{R}^n$ and taking values from $\mathbf{R}\cup\{+\infty\}$ is lower semicontinuous if and only if its epigraph is closed (e.g., due to its emptiness).*

We shall not prove this statement, same as most of other statements in this Section; the reader definitely is able to restore (very simple) proofs we are skipping.

An immediate consequence of the latter proposition is as follows:

Corollary 2.6.1 *The upper bound*

$$f(x) = \sup_{\alpha\in\mathcal{A}} f_\alpha(x)$$

of arbitrary family of lower semicontinuous functions is lower semicontinuous.

[from now till the end of the Section, if the opposite is not explicitly stated, "a function" means "a function defined on the entire $\mathbf{R}^n$ and taking values in $\mathbf{R}\cup\{+\infty\}$"]

Indeed, the epigraph of the upper bound is the intersection of the epigraphs of the functions forming the bound, and the intersection of closed sets always is closed.

Now let us look at *convex* lower semicontinuous functions; according to our general convention, "convex" means "satisfying the convexity inequality and finite at least at one point", or, which is the same, "with convex nonempty epigraph"; and as we just have seen, "lower semicontinuous" means "with closed epigraph". Thus, we are interested in functions with closed convex nonempty epigraphs; to save words, let us call these functions proper.

What we are about to do is to translate to the functional language several constructions and results related to convex sets. In the usual life, a translation (e.g. of poetry) typically results in something less rich than the original; in contrast to this, in mathematics this is a powerful source of new ideas and constructions.

"Outer description" of a proper function. We know that a closed convex set is intersection of closed half-spaces. What does this fact imply when the set is the epigraph of a proper function f? First of all, note that the epigraph is not a completely arbitrary convex set: it has a recessive direction $e = (1, 0)$ – the basic orth of the t-axis in the space of variables $t \in \mathbf{R}, x \in \mathbf{R}^n$ where the epigraph lives. This direction, of course, should be recessive for every closed half-space

$$(*)\quad \Pi = \{(t, x) : \alpha t \geq d^T x - a\} \quad [|\alpha| + |d| > 0]$$

containing $\mathrm{Epi}(f)$ (note that what is written in the right hand side of the latter relation, is one of many universal forms of writing down a general nonstrict linear inequality in the space where the epigraph lives; this is the form the most convenient for us now). Thus, e should be a recessive direction of $\Pi \supset \mathrm{Epi}(f)$; as it is immediately seen, recessivity of e for Π means exactly that $\alpha \geq 0$. Thus, speaking about closed half-spaces containing $\mathrm{Epi}(f)$, we in fact are considering some of the half-spaces (*) with $\alpha \geq 0$.

Now, there are two essentially different possibilities for α to be nonnegative – (A) to be positive, and (B) to be zero. In the case of (B) the boundary hyperplane of Π is "vertical" – it is parallel to e, and in fact it "bounds" only x – Π is comprised of all pairs (t, x) with x belonging to certain half-space in the x-subspace and t being arbitrary real. These "vertical" subspaces will be of no interest for us.

The half-spaces which indeed are of interest for us are the "nonvertical" ones: those given by the case (A), i.e., with $\alpha > 0$. For a non-vertical half-space Π, we always can divide the inequality defining Π by α and to make $\alpha = 1$. Thus, a "nonvertical" candidate to the role of a closed half-space containing $\mathrm{Epi}(f)$ always can be written down as

$$(**)\quad \Pi = \{(t, x) : t \geq d^T x - a\},$$

i.e., *can be represented as the epigraph of an affine function of x.*

Now, when such a candidate indeed is a half-space containing $\mathrm{Epi}(f)$? The answer is clear: it is the case *if and only if the affine function $d^Tx - a$ everywhere in $\mathbf{R}^n$ is $\le f(\cdot)$* – as we shall say, "is an affine minorant of f"; indeed, the smaller is the epigraph, the larger is the function. *If we knew that* $\mathrm{Epi}(f)$ – which definitely is the intersection of all closed half-spaces containing $\mathrm{Epi}(f)$ – *is in fact the intersection of already nonvertical closed half-spaces containing* $\mathrm{Epi}(f)$, or, which is the same, *the intersection of the epigraphs of all affine minorants of f*, we would be able to get a nice and nontrivial result:

> (!) *a proper convex function is the upper bound of affine functions – all its affine minorants.*

(indeed, we already know that it is the same – to say that a function is an upper bound of certain family of functions, and to say that the epigraph of the function is the intersection of the epigraphs of the functions of the family).

(!) indeed is true:

Proposition 2.6.2 *A proper convex function f is the upper bound of all its affine minorants. Moreover, at every point $\bar{x} \in \mathrm{rint\ Dom}\, f$ from the relative interior of the domain f f is even not the upper bound, but simply the maximum of its minorants: there exists an affine function $f_{\bar{x}}(x)$ which is $\le f(x)$ everywhere in $\mathbf{R}^n$ and is equal to f at $x = \bar{x}$.*

Proof. I. We start with the "Moreover" part of the statement; this is the key to the entire statement. Thus, we are about to prove that if $\bar{x} \in \mathrm{rint\ Dom}\, f$, then there exists an affine function $f_{\bar{x}}(x)$ which is everywhere $\le f(x)$, and at $x = \bar{x}$ the inequality becomes an equality.

I.1^0 First of all, we easily can reduce the situation to the one when $\mathrm{Dom}\, f$ is full-dimensional. Indeed, by shifting f we may make the affine span $\mathrm{Aff}(\mathrm{Dom}\, f)$ of the domain of f to be a linear subspace L in $\mathbf{R}^n$; restricting f onto this linear subspace, we clearly get a proper function on L. If we believe that our statement is true for the case when the domain of f is full-dimensional, we can conclude that there exists an affine function

$$d^Tx - a \quad [x \in L]$$

on L ($d \in L$) such that

$$f(x) \ge d^Tx - a \quad \forall x \in L; f(\bar{x}) = d^T\bar{x} - a.$$

The affine function we get clearly can be extended, by the same formula, from L on the entire $\mathbf{R}^n$ and is a minorant of f on the entire $\mathbf{R}^n$ – outside of $L \supset \mathrm{Dom}\, f$ f simply is $+\infty$! This minorant on $\mathbf{R}^n$ is exactly what we need.

I.2^0. Now let us prove that our statement is valid when $\mathrm{Dom}\, f$ is full-dimensional, so that $\bar{x}$ is an interior point of the domain of f. Let us look at the point $y = (f(\bar{x}), \bar{x})$. This is a point from the epigraph of f, and we claim that it is a point from the relative boundary of the epigraph. Indeed, if y were a relative interior point of $\mathrm{Epi}(f)$, then, taking $y' = y + e$, we would get a segment $[y', y]$ contained in $\mathrm{Epi}(f)$; since the endpoint y of the segment is assumed to be relative interior for $\mathrm{Epi}(f)$, we could extend this segment a little through this endpoint, not leaving $\mathrm{Epi}(f)$; but this clearly is impossible, since the t-coordinate of the new endpoint would be $< f(\bar{x})$, and the x-component of it still would be $\bar{x}$.

Thus, y is a point from the relative boundary of $\mathrm{Epi}(f)$. Now w claim that y' is an interior point of $\mathrm{Epi}(f)$. This is immediate: we know from Theorem 2.4.1 that f is continuous at $\bar{x}$, so that there exists a neighbourhood U of $\bar{x}$ in $\mathrm{Aff}(\mathrm{Dom}\, f) = \mathbf{R}^n$ such that $f(x) \le f(\bar{x}+0.5)$ whenever $x \in U$, or, in other words, the set

$$V = \{(t, x) : x \in U, t > f(\bar{x}) + 0.5\}$$

is contained in $\mathrm{Epi}(f)$; but this set clearly contains a neighbourhood of y' in $\mathbf{R}^{n+1}$.

Now let us look at the supporting linear form to $\mathrm{Epi}(f)$ at the point y of the relative boundary of $\mathrm{Epi}(f)$. This form gives us a linear inequality on $\mathbf{R}^{n+1}$ which is satisfied everywhere on $\mathrm{Epi}(f)$ and becomes equality at y; besides this, the inequality is not equality identically on $\mathrm{Epi}(f)$, it is strict somewhere on $\mathrm{Epi}(f)$. Without loss of generality we may assume that the inequality is of the form

$$(+) \quad \alpha t \geq d^T x - a.$$

Now, since our inequality is satisfied at $y' = y + e$ and becomes equality at $(t, x) = y$, α should be ≥ 0; it cannot be 0, since in the latter case the inequality in question would be equality also at $y' \in \mathrm{int}\,\mathrm{Epi}(f)$. But a linear inequality which is satisfied at a convex set and is *equality* at an *interior* point of the set is trivial – coming from the zero linear form (this is exactly the statement that a linear form attaining its minimum on a convex set at a point from the relative interior of the set is constant on the set and on its affine hull).

Thus, inequality $(+)$ which is satisfied on $\mathrm{Epi}(f)$ and becomes equality at y is an inequality with $\alpha > 0$. Let us divide both sides of the inequality by α; we get a new inequality of the form

$$(\&) \quad t \geq d^T x - a$$

(We keep the same notation for the right hand side coefficients – we never will come back to the old coefficients); this inequality is valid on $\mathrm{Epi}(f)$ and is equality at $y = (f(\bar{x}), \bar{x})$. Since the inequality is valid on $\mathrm{Epi}(f)$, it is valid at every pair (t, x) with $x \in \mathrm{Dom}\, f$ and $t = f(x)$:

$$(\#) \quad f(x) \geq d^T x - a \quad \forall x \in \mathrm{Dom}\, f;$$

so that the right hand side is an affine minorant of f on $\mathrm{Dom}\, f$ and therefore – on $\mathbf{R}^n$ ($f = +\infty$ outside $\mathrm{Dom}\, f$!). It remains to note that $(\#)$ is equality at $\bar{x}$, since $(\&)$ is equality at y. □

II. We have proved that if $\mathcal{F}$ if the set of all affine functions which are minorants of f, then the function

$$\bar{f}(x) = \sup_{\phi \in \mathcal{F}} \phi(x)$$

is equal to f on $\mathrm{rint}\,\mathrm{Dom}\, f$ (and at x from the latter set in fact sup in the right hand side can be replaced with max); to complete the proof of the Proposition, we should prove that $\bar{f}$ is equal to f also outside $\mathrm{rint}\,\mathrm{Dom}\, f$.

II.1^0. Let us first prove that $\bar{f}$ is equal to f outside $\mathrm{cl}\,\mathrm{Dom}\, f$, or. which is the same, prove that $\bar{f}(x) = +\infty$ outside $\mathrm{cl}\,\mathrm{Dom}\, f$. This is easy: is $\bar{x}$ is a point outside $\mathrm{cl}\,\mathrm{Dom}\, f$, it can be strongly separated from $\mathrm{Dom}\, f$, see Separation Theorem (ii) (Theorem 1.2.9). Thus, there exists $z \in \mathbf{R}^n$ such that

$$z^T \bar{x} \geq z^T x + \zeta \quad \forall x \in \mathrm{Dom}\, f \quad [\zeta > 0]. \tag{2.6.1}$$

Besides this, we already know that there exists at least one affine minorant of f, or, which is the same, there exist a and d such that

$$f(x) \geq d^T x - a \quad \forall x \in \mathrm{Dom}\, f. \tag{2.6.2}$$

Let us add to (2.6.2) inequality (2.6.1) multiplied by positive weight λ; we get

$$f(x) \geq \phi_\lambda(x) \equiv (d + \lambda z)^T x + [\lambda\zeta - a - \lambda z^T \bar{x}] \quad \forall x \in \mathrm{Dom}\, f.$$

This inequality clearly says that $\phi_\lambda(\cdot)$ is an affine minorant of f on $\mathbf{R}^n$ for every $\lambda > 0$. The value of this minorant at $x = \bar{x}$ is equal to $d^T \bar{x} - a + \lambda\zeta$ and therefore it goes to $+\infty$

as $\lambda \to +\infty$. We see that the upper bound of affine minorants of f at $\bar{x}$ indeed is $+\infty$, as claimed.

II.2^0. Thus, we know that the upper bound $\bar{f}$ of all affine minorants of f is equal to f everywhere on the relative interior of Dom f and everywhere outside the closure of Dom f; all we should prove that this equality is also valid at the points of the relative boundary of Dom f. Let $\bar{x}$ be such a point. There is nothing to prove if $\bar{f}(\bar{x}) = +\infty$, since by construction $\bar{f}$ is everywhere $\le f$. Thus, we should prove that if $\bar{f}(\bar{x}) = c < \infty$, then $f(\bar{x}) = c$. Since $\bar{f} \le f$ everywhere, to prove that $f(\bar{x}) = c$ is the same as to prove that $f(\bar{x}) \le c$. This is immediately given by lower semicontinuity of f: let us choose $x' \in$ rint Dom f and look what happens along a sequence of points $x_i \in [x', \bar{x})$ converging to $\bar{x}$. All the points of this sequence are relative interior points of Dom f (Lemma 1.1.1), and consequently

$$f(x_i) = \bar{f}(x_i).$$

Now, $x_i = (1 - \lambda_i)\bar{x} + \lambda_i x'$ with $\lambda_i \to +0$ as $i \to \infty$; since $\bar{f}$ clearly is convex (as the upper bound of a family of affine and therefore convex functions), we have

$$\bar{f}(x_i) \le (1 - \lambda_i)\bar{f}(\bar{x}) + \lambda_i \bar{f}(x').$$

Putting things together, we get

$$f(x_i) \le (1 - \lambda_i)\bar{f}(\bar{x}) + \lambda_i f(x');$$

as $i \to \infty$, $x_i \to \bar{x}$, and the right hand side in our inequality converges to $\bar{f}(\bar{x}) = c$; since f is lower semicontinuous, we get $f(\bar{x}) \le c$. ∎

We see why "translation of mathematical facts from one mathematical language to another" – in our case, from the language of convex sets to the language of convex functions – may be fruitful: because we invest a lot into the process rather than run it mechanically.

Closure of a convex function. We got a nice result on the "outer description" of a *proper* convex function: it is the upper bound of a family of affine functions. Note that, vice versa, the upper bound of every family of affine functions is a proper function, provided that this upper bound is finite at least at one point (indeed, as we know from Section 2.2.1, upper bound of every family of convex functions is convex, provided that it is finite at least at one point; and Corollary 2.6.1 says that upper bound of lower semicontinuous functions (e.g., affine ones – they are even continuous) is lower semicontinuous).

Now, what to do with a convex function which is not lower semicontinuous? The similar question about convex sets – what to do with a convex set which is not closed – can be resolved very simply: we can pass from the set to its closure and thus get a "normal" object which is very "close" to the original one: the "main part" of the original set – its relative interior – remains unchanged, and the "correction" adds to the set something relatively small – the relative boundary. The same approach works for convex functions: if a convex function f is not proper (i.e., its epigraph, being convex and nonempty, is not closed), we can "correct" the function – replace it with a new function with the epigraph being the closure of Epi(f). To justify this approach, we, of course, should be sure that the closure of the epigraph of a convex function is also an epigraph of such a function. This indeed is the case, and to see it, it suffices to note that a set G in $\mathbf{R}^{n+1}$ is the epigraph of a function taking values in $\mathbf{R} \cup \{+\infty\}$ if and only if the intersection of G with every vertical line $\{x = \text{const}, t \in \mathbf{R}\}$ is either empty, or is a closed ray of the form $\{x = \text{const}, t \ge \bar{t} > -\infty\}$. Now, it is absolutely evident that if G is the closure of the epigraph of a function f, that its intersection with a vertical line is either empty, or is a closed ray, or is the entire line (the last case indeed can take place – look at the closure of the epigraph of the function equal to $-\frac{1}{x}$ for $x > 0$ and $+\infty$ for $x \le 0$). We see that in order to justify our idea of "proper correction" of a convex function we should prove that if f is convex, then the last of the indicated three cases –

the intersection of $\mathrm{cl}\,\mathrm{Epi}(f)$ with a vertical line is the entire line – never occurs. This fact evidently is a corollary of the following simple

Proposition 2.6.3 *A convex function is below bounded on every bounded subset of* $\mathbf{R}^n$.

Proof. Without loss of generality we may assume that the domain of the function f is full-dimensional and that 0 is the interior point of the domain. According to Theorem 2.4.1, there exists a neighbourhood U of the origin – which can be thought of to be a centered at the origin ball of some radius $r > 0$ – where f is bounded from above by some C. Now, if $R > 0$ is arbitrary and x is an arbitrary point with $|x| \le R$, then the point

$$y = -\frac{r}{R}x$$

belongs to U, and we have

$$0 = \frac{r}{r+R}x + \frac{R}{r+R}y;$$

since f is convex, we conclude that

$$f(0) \le \frac{r}{r+R}f(x) + \frac{R}{r+R}f(y) \le \frac{r}{r+R}f(x) + \frac{R}{r+R}c,$$

and we get the lower bound

$$f(x) \ge \frac{r+R}{r}f(0) - \frac{r}{R}c$$

for the values of f in the centered at 0 ball of radius R. ∎

Thus, we conclude that the closure of the epigraph of a convex function f is the epigraph of certain function, let it be called *the closure* $\mathrm{cl}\,f$ *of* f. Of course, this latter function is convex (its epigraph is convex – it is the closure of a convex set), and since its epigraph is closed, $\mathrm{cl}\,f$ is proper. The following statement gives direct description of $\mathrm{cl}\,f$ in terms of f:

Proposition 2.6.4 *Let f be a convex function and $\mathrm{cl}\,f$ be its closure. Then*

(i) *For every x one has*

$$\mathrm{cl}\,f(x) = \lim_{r\to+0}\inf_{x':\|x'-x\|_2\le r} f(x').$$

In particular,

$$f(x) \ge \mathrm{cl}\,f(x)$$

for all x, and

$$f(x) = \mathrm{cl}\,f(x)$$

whenever $x \in \mathrm{rint}\,\mathrm{Dom}\,f$, same as whenever $x \notin \mathrm{cl}\,\mathrm{Dom}\,f$.

Thus, the "correction" $f \mapsto \mathrm{cl}\,f$ may vary f only at the points from the relative boundary of $\mathrm{Dom}\,f$,

$$\mathrm{Dom}\,f \subset \mathrm{Dom}\,\mathrm{cl}\,f \subset \mathrm{cl}\,\mathrm{Dom}\,f,$$

whence also

$$\mathrm{rint}\,\mathrm{Dom}\,f = \mathrm{rint}\,\mathrm{Dom}\,\mathrm{cl}\,f.$$

(ii) *The family of affine minorants of $\mathrm{cl}\,f$ is exactly the family of affine minorants of f, so that*

$$\mathrm{cl}\,f(x) = \sup\{\phi(x) : \phi \text{ is an affine minorant of } f\},$$

and the sup *in the right hand side can be replaced with* max *whenever $x \in \mathrm{rint}\,\mathrm{Dom}\,\mathrm{cl}\,f = \mathrm{rint}\,\mathrm{Dom}\,f$.*

["so that" comes from the fact that $\mathrm{cl}\,f$ is proper and is therefore the upper bound of its affine minorants]

2.6.2 Subgradients

Let f be a convex function, and let $x \in \mathrm{Dom}\, f$. It may happen that there exists an affine minorant $d^T x - a$ of f which coincides with f at x:

$$f(y) \geq d^T y - a \quad \forall y, \quad f(x) = d^T x - a.$$

From the equality in the latter relation we get $a = d^T x - f(x)$, and substituting this representation of a into the first inequality, we get

$$f(y) \geq f(x) + d^T(y - x) \quad \forall y. \tag{2.6.3}$$

Thus, if f admits an affine minorant which is exact at x, then there exists d which gives rise to inequality (2.6.3). Vice versa, if d is such that (2.6.3) takes place, then the right hand side of (2.6.3), regarded as a function of y, is an affine minorant of f which is exact at x.

Now note that (2.6.3) expresses certain property of a vector d. A vector satisfying, for a given x, this property – i.e., the slope of an exact at x affine minorant of f – is called a *subgradient* of f at x, and the set of all subgradients of f at x is denoted $\partial f(x)$.

Subgradients of convex functions play important role in the theory and numerical methods of Convex Programming – they are quite reasonable surrogates of gradients. The most elementary properties of the subgradients are summarized in the following statement:

Proposition 2.6.5 *Let f be a convex function and x be a point from* Dom f. *Then*

(i) *$\partial f(x)$ is a closed convex set which for sure is nonempty when $x \in$* rint Dom f

(ii) *If $x \in$* int Dom f *and f is differentiable at x, then $\partial f(x)$ is the singleton comprised of the usual gradient of f at x.*

Proof. (i): Closedness and convexity of $\partial f(x)$ are evident – (2.6.3) is an infinite system of nonstrict linear inequalities with respect to d, the inequalities being indexed by $y \in \mathbf{R}^n$. Nonemptiness of $\partial f(x)$ for the case when $x \in$ rint Dom f – this is the most important fact about the subgradients – is readily given by our preceding results. Indeed, we should prove that if $x \in$ rint Dom f, then there exists an affine minorant of f which is exact at x. But this is an immediate consequence of Proposition 2.6.4: part (i) of the proposition says that there exists an affine minorant of f which is equal to cl $f(x)$ at the point x, and part (i) says that $f(x) = \mathrm{cl}\, f(x)$.

(ii): If $x \in$ int Dom f and f is differentiable at x, then $\nabla f(x) \in \partial f(x)$ by the Gradient Inequality. To prove that in the case in question $\nabla f(x)$ is the only subgradient of f at x, note that if $d \in \partial f(x)$, then, by definition,

$$f(y) - f(x) \geq d^T(y - x) \quad \forall y$$

Substituting $y - x = th$, h being a fixed direction and t being > 0, dividing both sides of the resulting inequality by t and passing to limit as $t \to +0$, we get

$$h^T \nabla f(x) \geq h^T d.$$

This inequality should be valid for all h, which is possible if and only if $d = \nabla f(x)$. ∎

Proposition 2.6.5 explains why subgradients are good surrogates of gradients: at a point where gradient exists, it is the only subgradient, but, in contrast to the gradient, a subgradient exists basically everywhere (for sure in the relative interior of the domain of the function). For example, let us look at the simple function

$$f(x) = |x|$$

on the axis. It is, of course, convex (as maximum of two linear forms x and $-x$). Whenever $x \neq 0$, f is differentiable at x with the derivative $+1$ for $x > 0$ and -1 for $x < 0$. At the point

$x = 0$ f is not differentiable; nevertheless, it must have subgradients at this point (since 0 is an interior point of the domain of the function). And indeed, it is immediately seen that the subgradients of $|x|$ at $x = 0$ are exactly the reals from the segment $[-1, 1]$. Thus,

$$\partial|x| = \begin{cases} \{-1\}, & x < 0 \\ [-1, 1], & x = 0 \\ \{+1\}, & x > 0 \end{cases}.$$

Note also that if x is a relative boundary point of the domain of a convex function, even a "good" one, the set of subgradients of f at x may be empty, as it is the case with the function

$$f(y) = \begin{cases} -\sqrt{y}, & y \geq 0 \\ +\infty, & y < 0 \end{cases};$$

it is clear that there is no non-vertical supporting line to the epigraph of the function at the point $(0, f(0))$, and, consequently, there is no affine minorant of the function which is exact at $x = 0$.

A significant – and important – part of Convex Analysis deals with *subgradient calculus* – with the rules for computing subgradients of "composite" functions, like sums, superpositions, maxima, etc., given subgradients of the operands. These rules extend onto nonsmooth convex case the standard Calculus rules and are very nice and instructive; the related considerations, however, are beyond our scope.

2.6.3 Legendre transformation

Let f be a convex function. We know that f "basically" is the upper bound of all its affine minorants; this is exactly the case when f is proper, otherwise the corresponding equality takes place everywhere except, perhaps, some points from the relative boundary of Dom f. Now, when an affine function $d^T x - a$ is an affine minorant of f? It is the case if and only if

$$f(x) \geq d^T x - a$$

for all x or, which is the same, if and only if

$$a \geq d^T x - f(x)$$

for all x. We see that if the slope d of an affine function $d^T x - a$ is fixed, then in order for the function to be a minorant of f we should have

$$a \geq \sup_{x \in \mathbf{R}^n} [d^T x - f(x)].$$

The supremum in the right hand side of the latter relation is certain function of d; this function is called the *Legendre transformation* of f and is denoted f^*:

$$f^*(d) = \sup_{x \in \mathbf{R}^n} [d^T x - f(x)].$$

Geometrically, the Legendre transformation answers the following question: given a slope d of an affine function, i.e., given the hyperplane $t = d^T x$ in $\mathbf{R}^{n+1}$, what is the minimal "shift down" of the hyperplane which places it below the graph of f?

From the definition of the Legendre transformation it follows that this is a proper function. Indeed, we loose nothing when replacing $\sup_{x \in \mathbf{R}^n} [d^T x - f(x)]$ by $\sup_{x \in \mathrm{Dom}\, f} [d^T x - f(x)]$, so that the Legendre transformation is the upper bound of a family of affine functions. Since this bound is finite at least at one point (namely, at every d coming form affine minorant of f; we know that such a minorant exists), it is a convex lower semicontinuous function, as claimed.

The most elementary (and the most fundamental) fact about the Legendre transformation is its symmetry:

Proposition 2.6.6 *Let f be a convex function. Then twice taken Legendre transformation of f is the closure $\mathrm{cl}\, f$ of f:*

$$(f^*)^* = \mathrm{cl}\, f.$$

In particular, if f is proper, then it is the Legendre transformation of its Legendre transformation (which also is proper).

Proof is immediate. The Legendre transformation of f^* at the point x is, by definition,

$$\sup_{d\in\mathbf{R}^n}[x^Td - f^*(d)] = \sup_{d\in\mathbf{R}^n, a\geq f^*(d)}[d^Tx - a];$$

the second sup here is exactly the supremum of all affine minorants of f (this is the origin of the Legendre transformation: $a \geq f^*(d)$ if and only if the affine form $d^Tx - a$ is a minorant of f). And we already know that the upper bound of all affine minorants of f is the closure of f. ∎

The Legendre transformation is a very powerful tool – this is a "global" transformation, so that *local* properties of f^* correspond to *global* properties of f. For example,

- $d = 0$ belongs to the domain of f^* if and only if f is below bounded, and if it is the case, then $f^*(0) = -\inf f$;
- if f is proper, then the subgradients of f^* at $d = 0$ are exactly the minimizers of f on $\mathbf{R}^n$;
- Dom f^* is the entire $\mathbf{R}^n$ if and only if $f(x)$ grows, as $\| x \|_2 \to \infty$, faster than $\| x \|_2$: there exists a function $r(t) \to \infty$, as $t \to \infty$ such that

$$f(x) \geq r(\| x \|_2) \quad \forall x,$$

etc. Thus, whenever we can compute explicitly the Legendre transformation of f, we get a lot of "global" information on f. Unfortunately, the more detailed investigation of the properties of Legendre transformation is beyond our scope; I simply list several simple facts and examples:

- From the definition of Legendre transformation,

$$f(x) + f^*(d) \geq x^Td \quad \forall x, d.$$

Specifying here f and f^*, we get certain inequality, e.g., the following one:
[Young's Inequality] *if p and q are positive reals such that $\frac{1}{p} + \frac{1}{q} = 1$, then*

$$\frac{|x|^p}{p} + \frac{|d|^q}{q} \geq xd \quad \forall x, d \in \mathbf{R}$$

(indeed, as it is immediately seen, the Legendre transformation of the function $|x|^p/p$ is $|d|^q/q$)

Consequences. Very simple-looking Young's inequality gives rise to a very nice and useful *Hölder inequality:*
Let $1 \leq p \leq \infty$ and let q be such $\frac{1}{p} + \frac{1}{q} = 1$ ($p = 1 \Rightarrow q = \infty$, $p = \infty \Rightarrow q = 1$). For every two vectors $x, y \in \mathbf{R}^n$ one has

$$\sum_{i=1}^n |x_iy_i| \leq \| x \|_p \| y \|_q \tag{2.6.4}$$

Indeed, there is nothing to prove if p or q is ∞ – if it is the case, the inequality becomes the evident relation

$$\sum_i |x_i y_i| \le (\max_i |x_i|)(\sum_i |y_i|).$$

Now let $1 < p < \infty$, so that also $1 < q < \infty$. In this case we should prove that

$$\sum_i |x_i y_i| \le (\sum_i |x_i|^p)^{1/p} (\sum_i |y_i|^q)^{1/q}.$$

There is nothing to prove if one of the factors in the right hand side vanishes; thus, we can assume that $x \neq 0$ and $y \neq 0$. Now, both sides of the inequality are of homogeneity degree 1 with respect to x (when we multiply x by t, both sides are multiplied by $|t|$), and similarly with respect to y. Multiplying x and y by appropriate reals, we can make both factors in the right hand side equal to 1: $\| x \|_p = \| y \|_p = 1$. Now we should prove that under this normalization the left hand side in the inequality is ≤ 1, which is immediately given by the Young inequality:

$$\sum_i |x_i y_i| \le \sum_i [|x_i|^p/p + |y_i|^q/q] = 1/p + 1/q = 1.$$

Note that the Hölder inequality says that

$$|x^T y| \le \| x \|_p \| y \|_q; \tag{2.6.5}$$

when $p = q = 2$, we get the Cauchy inequality. Now, inequality (2.6.5) is *precise* in the sense that for every x there exists y with $\| y \|_q = 1$ such that

$$x^T y = \| x \|_p \quad [= \| x \|_p \| y \|_q];$$

it suffices to take

$$y_i = \| x \|_p^{1-p} |x_i|^{p-1} \mathrm{sign}(x_i)$$

(here $x \neq 0$; the case of $x = 0$ is trivial – here y can be an arbitrary vector with $\| y \|_q = 1$).

Combining our observations, we come to an extremely important, although simple, fact:

$$\| x \|_p = \max\{y^T x : \| y \|_q \le 1\} \quad [\frac{1}{p} + \frac{1}{q} = 1]. \tag{2.6.6}$$

It follows, in particular, that $\| x \|_p$ is convex (as an upper bound of a family of linear forms), whence

$$\| x' + x'' \|_p = 2 \| \frac{1}{2} x' + \frac{1}{2} x'' \|_p \le 2(\| x' \|_p /2 + \| x'' \|_p /2) = \| x' \|_p + \| x'' \|_p;$$

this is nothing but the triangle inequality. Thus, $\| x \|_p$ satisfies the triangle inequality; it clearly possesses two other characteristic properties of a norm – positivity and homogeneity. Consequently, $\| \cdot \|_p$ is a norm – the fact that we announced twice and have finally proven now.

- The Legendre transformation of the function

$$f(x) \equiv -a$$

is the function which is equal to a at the origin and is $+\infty$ outside the origin; similarly, the Legendre transformation of an affine function $\bar{d}^T x - a$ is equal to a at $d = \bar{d}$ and is $+\infty$ when $d \neq \bar{d}$;

- The Legendre transformation of the strictly convex quadratic form

$$f(x) = \frac{1}{2}x^T Ax$$

(A is positive definite symmetric matrix) is the quadratic form

$$f^*(d) = \frac{1}{2}d^T A^{-1} d$$

- The Legendre transformation of the Euclidean norm

$$f(x) = \| x \|_2$$

is the function which is equal to 0 in the closed unit ball centered at the origin and is $+\infty$ outside the ball.

The latter example is a particular case of the following statement:

Let $\| x \|$ be a norm on $\mathbf{R}^n$, and let

$$\| d \|_* = \sup\{d^T x : \| x \| \leq 1\}$$

be the conjugate to $\| \cdot \|$ norm.

> ***Exercise 2.1*** *Prove that $\| \cdot \|_*$ is a norm, and that the norm conjugate to $\| \cdot \|_*$ is the original norm $\| \cdot \|$.*
> Hint: Observe that the unit ball of $\| \cdot \|_*$ is exactly the polar of the unit ball of $\| \cdot \|$.

The Legendre transformation of $\| x \|$ is the characteristic function of the unit ball of the conjugate norm, i.e., is the function of d equal to 0 when $\| d \|_* \leq 1$ and is $+\infty$ otherwise.

For example, (2.6.6) says that the norm conjugate to $\| \cdot \|_p$, $1 \leq p \leq \infty$, is $\| \cdot \|_q$, $1/p + 1/q = 1$; consequently, the Legendre transformation of p-norm is the characteristic function of the unit $\| \cdot \|_q$-ball.

Lecture 3

Convex Programming, Lagrange Duality, Saddle Points

3.1 Mathematical Programming Program

A (constrained) Mathematical Programming program is a problem as follows:

$$\text{(P)} \quad \min\{f(x) : x \in X, \quad g(x) \equiv (g_1(x), ..., g_m(x)) \leq 0, \ h(x) \equiv (h_1(x), ..., h_k(x)) = 0\}. \tag{3.1.1}$$

The standard terminology related to (3.1.1) is:

- [domain] X is called the *domain* of the problem
- [objective] f is called the *objective*
- [constraints] g_i, $i = 1, ..., m$, are called the (functional) *inequality constraints*; h_j, $j = 1, ..., k$, are called the *equality constraints*[1)]

In the sequel, if the opposite is not explicitly stated, it always is assumed that the objective and the constraints are well-defined on X.

- [feasible solution] a point $x \in \mathbf{R}^n$ is called a *feasible solution* to (3.1.1), if $x \in X$, $g_i(x) \leq 0$, $i = 1, ..., m$, and $h_j(x) = 0$, $j = 1, ..., k$, i.e., if x satisfies all restrictions imposed by the formulation of the problem
  - [feasible set] the set of all feasible solutions is called the *feasible set* of the problem
  - [feasible problem] a problem with a nonempty feasible set (i.e., the one which admits feasible solutions) is called *feasible* (or consistent)
  - [active constraints] an inequality constraint $g_i(\cdot) \leq 0$ is called *active at a given feasible solution* x, if this constraint is satisfied at the point as an equality rather than strict inequality, i.e., if
    $$g_i(x) = 0.$$
    A equality constraint $h_i(x) = 0$ by definition is active at every feasible solution x.
- [optimal value] the quantity
  $$f^* = \begin{cases} \inf\limits_{x \in X : g(x) \leq 0, h(x) = 0} f(x), & \text{the problem is feasible} \\ +\infty, & \text{the problem is infeasible} \end{cases}$$
  is called *the optimal value* of the problem

[1)] rigorously speaking, the constraints are not the functions g_i, h_j, but the relations $g_i(x) \leq 0$, $h_j(x) = 0$; in fact the word "constraints" is used in both these senses, and it is always clear what is meant. For example, saying that x satisfies the constraints, we mean the relations, and saying that the constraints are differentiable, we mean the functions

  - [below boundedness] the problem is called *below bounded*, if its optimal value is $> -\infty$, i.e., if the objective is below bounded on the feasible set

- [optimal solution] a point $x \in \mathbf{R}^n$ is called an *optimal solution* to (3.1.1), if x is feasible and $f(x) \leq f(x')$ for any other feasible solution, i.e., if

$$x \in \operatorname*{Argmin}_{x' \in X: g(x') \leq 0, h(x')=0} f(x')$$

  - [solvable problem] a problem is called *solvable*, if it admits optimal solutions
  - [optimal set] the set of all optimal solutions to a problem is called its *optimal set*

To solve the problem *exactly* means to find its optimal solution or to detect that no optimal solution exists.

3.2 Convex Programming program and Lagrange Duality Theorem

A Mathematical Programming program (P) is called *convex* (or *Convex Programming* program), if

- X is a *convex* subset of $\mathbf{R}^n$
- $f, g_1, ..., g_m$ are *real-valued convex* functions on X,
  and
- there are no equality constraints at all.

Note that instead of saying that there are no equality constraints, we could say that there are constraints of this type, but only *linear* ones; this latter case can be immediately reduced to the one without equality constraints by replacing $\mathbf{R}^n$ with the affine subspace given by the (linear) equality constraints.

3.2.1 Convex Theorem on Alternative

The simplest case of a convex program is, of course, a Linear Programming program – the one where $X = \mathbf{R}^n$ and the objective and all the constraints are linear. We already know what are optimality conditions for this particular case – they are given by the Linear Programming Duality Theorem. How did we get these conditions?

We started with the observation that the fact that a point x^* is an optimal solution can be expressed in terms of solvability/unsolvability of certain systems of inequalities: in our now terms, these systems are

$$x \in G,\ f(x) \leq c,\ g_j(x) \leq 0,\ j = 1, ..., m \tag{3.2.1}$$

and

$$x \in G,\ f(x) < c,\ g_j(x) \leq 0,\ j = 1, ..., m; \tag{3.2.2}$$

here c is a parameter. Optimality of x^* for the problem means exactly that for appropriately chosen c (this choice, of course, is $c = f(x^*)$) the first of these systems is solvable and x^* is its solution, while the second system is unsolvable. Given this trivial observation, we converted the "negative" part of it – the claim that (3.2.2) is unsolvable – into a positive statement, using the General Theorem on Alternative, and this gave us the LP Duality Theorem.

Now we are going to use the same approach. What we need is a "convex analogy" to the Theorem on Alternative – something like the latter statement, but for the case when the inequalities in question are given by convex functions rather than the linear ones (and, besides it, we have a "convex inclusion" $x \in X$).

It is easy to *guess* the result we need. How did we come to the formulation of the Theorem on Alternative? The question we were interested in was, basically, how to express in an affirmative manner

the fact that a system of linear inequalities has no solutions; to this end we observed that if we can combine, in a linear fashion, the inequalities of the system and get an obviously false inequality like $0 \leq -1$, then the system is unsolvable; this condition is certain affirmative statement with respect to the weights with which we are combining the original inequalities.

Now, the scheme of the above reasoning has nothing in common with linearity (and even convexity) of the inequalities in question. Indeed, consider *an arbitrary* inequality system of the type (3.2.2):

$$\begin{array}{rcl} (\mathrm{I}): & & \\ f(x) & < & c \\ g_j(x) & \leq & 0,\ j = 1, ..., m \\ x & \in & X; \end{array}$$

all we assume is that X is a nonempty subset in $\mathbf{R}^n$ and $f, g_1, ..., g_m$ are real-valued functions on X. It is absolutely evident that

if there exist nonnegative $\lambda_1, ..., \lambda_m$ such that the inequality

$$f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) < c \tag{3.2.3}$$

has no solutions in X, then (I) *also has no solutions.*

Indeed, a solution to (I) clearly is a solution to (3.2.3) – the latter inequality is nothing but a combination of the inequalities from (I) with the weights 1 (for the first inequality) and λ_j (for the remaining ones).

Now, what does it mean that (3.2.3) has no solutions? A necessary and sufficient condition for this is that the infimum of the left hand side of (3.2.3) in $x \in X$ is $\geq c$. Thus, we come to the following evident

Proposition 3.2.1 [Sufficient condition for insolvability of (I)] *Consider a system* (I) *with arbitrary data and assume that the system*

$$\begin{array}{rcl} (\mathrm{II}): & & \\ \inf\limits_{x \in X} \left[f(x) + \sum\limits_{j=1}^{m} \lambda_j g_j(x) \right] & \geq & c \\ \lambda_j & \geq & 0,\ j = 1, ..., m \end{array}$$

with unknowns $\lambda_1, ..., \lambda_m$ has a solution. Then (I) *is infeasible.*

Let us stress that this result is completely general; it does not require any assumptions on the entities involved.

The result we have obtained, unfortunately, does not help us: the actual power of the Theorem on Alternative (and the fact used to prove the Linear Programming Duality Theorem) is not the *sufficiency* of the condition of Proposition for infeasibility of (I), but the *necessity* of this condition. Justification of necessity of the condition in question has nothing in common with the evident reasoning which gives the sufficiency. The necessity in the linear case ($X = \mathbf{R}^n$, f, $g_1, ..., g_m$ are linear) can be established via the Homogeneous Farkas Lemma. Now we shall prove the necessity of the condition for the *convex* case, and already here we need some additional, although minor, assumptions; and in the general nonconvex case the condition in question simply is *not* necessary for infeasibility of (I) [and this is very bad – this is the reason why there exist difficult optimization problems which we do not know how to solve efficiently].

The just presented "preface" explains what we should do; now let us carry out our plan. We start with the aforementioned "minor regularity assumptions".

Definition 3.2.1 [Slater Condition] *Let $X \subset \mathbf{R}^n$ and $g_1, ..., g_m$ be real-valued functions on X. We say that these functions satisfy the Slater condition on X, if there exists x in the relative interior of X such that $g_j(x) < 0$, $j = 1, ..., m$.*

An inequality constrained program

$$(\mathrm{IC}) \qquad \min \{ f(x) : g_j(x) \leq 0,\ j = 1, ..., m,\ x \in X \}$$

($f, g_1, ..., g_m$ are real-valued functions on X) is called to satisfy the Slater condition, if $g_1, ..., g_m$ satisfy this condition on X.

Definition 3.2.2 [Relaxed Slater Condition] *Let $X \subset \mathbf{R}^n$ and $g_1, ..., g_m$ be real-valued functions on X. We say that these functions satisfy the Relaxed Slater condition on X, if, after appropriate reordering of $g_1, ..., g_m$, for properly selected $k \in \{0, 1, ..., m\}$ the functions $g_1, ..., g_k$ are affine, and there exists x in the relative interior of X such that $g_j(x) \leq 0$, $1 \leq j \leq k$, and $g_j(x) < 0$, $k+1 \leq j \leq m$.*

An inequality constrained problem (IC) *is called to satisfy the Relaxed Slater condition, if $g_1, ..., g_m$ satisfy this condition on X.*

Clearly, the validity of Slater condition implies the validity of the Relaxed Slater condition (in the latter, we should set $k = 0$). We are about to establish the following fundamental fact:

Theorem 3.2.1 [Convex Theorem on Alternative]
Let $X \subset \mathbf{R}^n$ be convex, let $f, g_1, ..., g_m$ be real-valued convex functions on X, and let $g_1, ..., g_m$ satisfy the Relaxed Slater condition on X. Then system (I) *is solvable if and only if system* (II) *is unsolvable.*

3.2.1.1 Cone constrained case

We shall obtain Theorem 3.2.1 as a particular case of *Convex Theorem on Alternative in cone constrained form* dealing with convex system (I) *in cone constrained form:*

$$f(x) < c,\ \overline{g}(x) := Ax - b \leq 0,\ \widehat{g}(x) \in -\mathbf{K},\ x \in X, \tag{3.2.4}$$

where

- X is a nonempty convex set in $\mathbf{R}^n$, and $f : X \to \mathbf{R}$ is convex,
- $\mathbf{K} \subset \mathbf{R}^\nu$ is a regular cone, regularity meaning that $\mathbf{K}$ is closed, convex, possesses a nonempty interior, and is pointed, that is $\mathbf{K} \cap [-\mathbf{K}] = \{0\}$
- $\widehat{g}(x) : Z \to \mathbf{R}^\nu$ is $\mathbf{K}$-*convex*, meaning that whenever $x, y \in X$ and $\lambda \in [0, 1]$, we have

$$\lambda\widehat{g}(x) + (1-\lambda)\widehat{g} - \widehat{g}(\lambda x + (1-\lambda)y) \in \mathbf{K}.$$

Note that in the simplest case of $\mathbf{K} = \mathbf{R}^\nu_+$ (nonnegative orthant is a regular cone!) $\mathbf{K}$-convexity means exactly that $\widehat{g}$ is a vector function with convex on X components.

Given a regular cone $\mathbf{K} \subset \mathbf{R}^\nu$, we can associate with it *K-inequality between vectors of $\mathbf{R}^\nu$*. saying that $a \in \mathbf{R}^\nu$ is $\mathbf{K}$-greater then or equal to $b \in \mathbf{R}^\nu$ (notation: $A \geq_{\mathbf{K}} b$, or, equivalently, $b \leq_{\mathbf{K}} a$) when $a - b \in \mathbf{K}$:

$$a \geq_{\mathbf{K}} b \Leftrightarrow b \leq_{\mathbf{K}} a \Leftrightarrow a - b \in \mathbf{K}$$

For example, when $\mathbf{K} = \mathbf{R}^\nu_+$ is nonnegative orthant, $\geq_{\mathbf{K}}$ is the standard coordinate-wise vector inequality $\geq$: $a \geq b$ means that every entry of a is greater than or equal to, in the standard arithmetic sense, the corresponding entry in b. $\mathbf{K}$- vector inequality possesses all algebraic properties of $\geq$:

1. it is a partial order on $\mathbf{R}^\nu$, meaning that the relation $a \geq_{\mathbf{K}} b$ is
   - reflexive: $a \geq_{\mathbf{K}} a$ for all a
   - anti-symmetric: $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} a$ if and only if $a = b$
   - transitive: is $a \geq_{\mathbf{K}} b$ and $b \geq_{\mathbf{K}} c$, then $a \geq_{\mathbf{K}} c$
2. is compatible with linear operations, meaning that
   - $\geq$-inequalities can be summed up: is $a \geq_{\mathbf{K}} b$ and $c \geq_{\mathbf{K}} d$, then $a + c \geq_{\mathbf{K}} b + d$
   - multiplied by nonnegative reals: if $a \geq_{\mathbf{K}} b$ and λ is a nonnegative real, then $\lambda a \geq_{\mathbf{K}} \lambda b$

3. is compatible with convergence, meaning that one can pass to sidewise limits in $\geq_{\mathbf{K}}$-inequality:
   - if $a_t \geq_{\mathbf{K}} b_t$, $t = 1, 2, ...$, and $a_t \to a$ and $b_t \to b$ as $t \to \infty$, then $a \geq_{\mathbf{K}} b$
4. gives rise to strict version $>_{\mathbf{K}}$ of $\geq_{\mathbf{K}}$-inequality: $a >_{\mathbf{K}} b$ (equivalently: $b <_{\mathbf{K}} a$) meaning that $a - b \in \operatorname{int} \mathbf{K}$. The strict $\mathbf{K}$-inequality possesses the basic properties of the coordinate-wise $>$, specifically
   - $>_{\mathbf{K}}$ is stable: if $a >_{\mathbf{K}} b$ and a', b' are close enough to a, b respectively, then $a' >_{\mathbf{K}} b'$
   - if $a >_{\mathbf{K}} b$, λ is a positive real, and $c \geq_{\mathbf{K}} d$, then $\lambda a >_{\mathbf{K}} \lambda b$ and $a + c >_{\mathbf{K}} b + d$

In summary, the arithmetics of $\leq_{\mathbf{K}}$ and $<_{\mathbf{K}}$ inequalities is completely similar to the one of the usual $\geq$ and $>$.

Exercise 3.1 *Verify the above claims*

Since 1990's, it was realized that as far as nonlinear Convex Optimization is concerned, it is extremely convenient to consider, along with the usual Mathematical Programming form

$$\min_{x \in X} \{g_0(x) : g(x) := [g_1(x); ...; g_m(x)] \leq 0, 1 \leq i \leq m, h_j(x) = 0, j \leq k\}$$
$$[X\text{: convex set; } g_i(x) : X \to \mathbf{R}, 0 \leq i \leq m\text{: convex; } h_j(x), j \leq k\text{: affine}]$$

of a convex problem, where nonlinearity "sits" in g_i's and/or non-polyhedrality of X, the *cone constrained form* where the nonlinearity "sits" in $\leq$ - the usual coordinate-wise $\leq$ is replaced with $\leq_{\mathbf{K}}$, $\mathbf{K}$ being a regular cone. The resulting *convex program in cone constrained form* reads

$$\min_{x \in X} \left\{ f(x) : \overline{g}(x) := Ax - b \leq 0, \underbrace{\widehat{g}(x) \leq_{\mathbf{K}} 0}_{\Leftrightarrow \widehat{g}(x) \in -\mathbf{K}} \right\} \tag{3.2.5}$$

where (cf. (3.2.4) X is convex set, $f : X \to \mathbf{R}$ is convex function, $\mathbf{K} \subset \mathbf{R}^\nu$ is a regular cone, and $\widehat{g} : X \to \mathbf{R}^\nu$ is $\mathbf{K}$-convex. Note that $\mathbf{K}$-convexity of $\widehat{g}$ is our new notation takes the quite common for us form

$$\forall (x, y \in X, \lambda \in [0, 1]) : \widehat{g}(\lambda x + (1 - \lambda) y)) \leq_{\mathbf{K}} \lambda \widehat{g}(x) + (1 - \lambda) \widehat{g}(y).$$

Note also that when $\mathbf{K}$ is nonnegative orthant, (3.2.5) recovers the Mathematical Programming form of a convex problem.

Exercise 3.2 *Let $\mathbf{K}$ be the cone of $m \times m$ positive semidefinite matrices in the space $\mathbf{S}^m$ of $m \times m$ symmetric matrices[2], and let $\widehat{g}(x) = xx^T : X := \mathbf{R}^{m \times n} \to \mathbf{S}^m$. Verify that $\widehat{g}$ is $\mathbf{K}$-convex.*

It turns out that with "cone constrained approach" to Convex Programming, we loose nothing when restricting ourselves with $X = \mathbf{R}^n$, linear $f(x)$, and affine $\widehat{g}(x)$; this specific version of (3.2.5) is called "conic problem" (to be considered in more detaile later). In our course, where convex problems are *not* the only subject, it makes sense to speak about a "less extreme" cone constrained form of a convex program, specifically, one presented in (3.2.5); we call problems of this form "convex problems in cone constrained form," reserving the words "conic problems" for problems (3.2.5) with $X = \mathbf{R}^n$, linear f and affine $\widehat{g}$.

[2] So far, we have assumed that $\mathbf{K}$ "lives" in some $\mathbf{R}^\nu$, while now we are speaking about something living in the space of symmetric matrices. Well, we can identify the latter linear space with $\mathbf{R}^\nu$, $\nu = \frac{m(m+1)}{2}$, by representing a symmetric matrix $[a_{ij}]$ by the collection of its diagonal and below-diagonal entries arranged into a column vector by writing down these entries, say, row by row: $[a_{ij}] \mapsto [a_{1,1}; a_{2,1}; a_{2,2}; ...; a_{m,1}; ...; a_{m,m}]$.

What we have done so far in this section can be naturally extended to the cone constrained case. Specifically, (in what follows $X \subset \mathbf{R}^n$ is a nonempty set, $\mathbf{K} \subset \mathbf{R}^\nu$ is a regular cone, f is a real-valued function on X, and $\widehat{g}(\cdot)$ is a mapping from X into $\mathbf{R}^\nu$)

1. Instead of feasibility/infeasibility of system (I) we can speak about feasibility/infeasibility of system of constraints

$$\text{(ConI)}: \quad \begin{array}{rcl} f(x) & < & c \\ \overline{g}(x) := Ax - b & \leq & 0 \\ \widehat{g}(x) & \leq_{\mathbf{K}} & 0 \ [\Leftrightarrow \widehat{g}(x) \in -\mathbf{K}] \\ x & \in & X \end{array}$$

in variables x. Denoting by $\mathbf{K}_*$ the cone dual to $\mathbf{K}$, a *sufficient* condition for infeasibility of (ConI) is solvability of system of constraints

$$\text{(ConII)}: \quad \begin{array}{rcl} \inf\limits_{x\in X}\left[f(x) + \overline{\lambda}^T\overline{g}(x) + \widehat{\lambda}^T\widehat{g}(x)\right] & \geq & c \\ \overline{\lambda} & \geq & 0 \\ \widehat{\lambda} & \geq_{\mathbf{K}_*} & 0 \ [\Leftrightarrow \widehat{\lambda} \in \mathbf{K}_*] \end{array}$$

in variables $\lambda = (\overline{\lambda}, \widehat{\lambda})$.

Indeed, given a feasible solution $\overline{\lambda}, \widehat{\lambda}$ to (ConII) and "aggregating" the constraints in (ConI) with weights $1, \overline{\lambda}, \widehat{\lambda}$ (i.e., taking sidewise inner products of constraints in (ConI) and weights and summing up the results), we arrive at the inequality

$$f(x) + \overline{\lambda}^T\overline{g}(x) + \widehat{\lambda}^T\widehat{g}(x) < c$$

which due to $\overline{\lambda} \geq 0$ and $\widehat{\lambda} \in \mathbf{K}_*$ is a consequence of (ConI) – it must be satisfied at every feasible solution to (ConI). On the other hand, the aggregated inequality contradicts the first constraint in (ConII), and we conclude that under the circumstances (ConI) is infeasible.

2. "Cone constrained" version of Slater/Relaxed Slater condition is as follows: we say that the system of constraints

$$\text{(S)} \qquad \overline{g}(x) := Ax - b \leq 0 \ \& \ \widehat{g}(x) \leq_{\mathbf{K}} 0$$

in variables x satisfies
— Slater condition on X, if there exists $\bar{x} \in \operatorname{rint} X$ such that $\overline{g}(\bar{x}) < 0$ and $\widehat{g}(\bar{x}) <_{\mathbf{K}} 0$ (i.e., $\widehat{g}(\bar{x}) \in -\operatorname{int}\mathbf{K}$),
— Relaxed Slater condition on X, if there exists $\bar{x} \in \operatorname{rint} X$ such that $\overline{g}(\bar{x}) \leq 0$ and $\widehat{g}(\bar{x}) <_{\mathbf{K}} 0$.
We say that optimization problem in cone constrained form – problem of the form

$$\text{(ConIC)} \qquad \min_{x\in X}\{f(x) : \overline{g}(x) := Ax - b \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0\}$$

satisfies Slater/Relaxed Slater condition, if the system of its constraints satisfies this condition on X.

Note that in the case of $\mathbf{K} = \mathbf{R}^\nu_+$, (ConI) and (ConII) become, respectively, (I) and (II), and the cone constrained versions of Slater/Relaxed Slater condition become the usual ones.

The cone constrained version of Theorem 3.2.1 reads as follows:

Theorem 3.2.2 [Convex Theorem on Alternative in cone constrained form]
Let $\mathbf{K} \subset \mathbf{R}^\nu$ be a regular cone, let $X \subset \mathbf{R}^n$ be convex, and let f be real-valued convex function on X, $\overline{g}(x) = Ax - b$ be affine, and $\widehat{g}(x) : X \to \mathbf{R}^\nu$ be $\mathbf{K}$-convex. Let also system (S) *satisfy Relaxed Slater condition on X. Then system* (ConI) *is solvable if and only if system* (ConII) *is unsolvable.*

Note: "In reality" (ConI) may have no polyhedral part $\overline{g}(x) := Ax - b \leq 0$ and/or no "general part" $\widehat{g}(x) \leq_{\mathbf{K}} 0$; absence of one or both of these parts leads to self-evident modifications in (ConII). To unify our forthcoming considerations, it is convenient to assume that both these parts are present. This assumption is for free: it is immediately seen that in our context, in absence of one or both of g-constraints in (ConI) we lose nothing when adding artificial polyhedral part $\overline{g}(x) := 0^T x - 1 \leq 0$ instead of missing polyhedral part, and/or artificial general part $\widehat{g}(x) := 0^T x - 1 \leq_{\mathbf{K}} 0$ with $\mathbf{K} = \mathbf{R}_+$ instead of missing general part. Thus, we lose nothing when assuming from the very beginning that both polyhedral and general parts are present.

It is immediately seen that Convex Theorem on Alternative (Theorem 3.2.1) is a special case of the latter Theorem corresponding to the case when $\mathbf{K}$ is a nonnegative orthant.

Proof of Theorem 3.2.2. The first part of the statement – "if (ConII) has a solution, then (ConI) has no solutions" – has been already verified. What we need is to prove the inverse statement. Thus, let us assume that (ConI) has no solutions, and let us prove that then (ConII) has a solution.

0^0. Without loss of generality we may assume that X is full-dimensional: $\operatorname{rint} X = \operatorname{int} X$ (indeed, otherwise we could replace our "universe" $\mathbf{R}^n$ with the affine span of X). Besides this, shifting f by constant, we can assume that $c = 0$. Thus, we are in the case where

$$
\begin{array}{ll}
\text{(ConI)}: & \\
& \begin{array}{rcl} f(x) & < & 0 \\ \overline{g}(x) := Ax - b & \leq & 0, \\ \widehat{g}(x) & \leq_{\mathbf{K}} & 0, \ [\Leftrightarrow \widehat{g}(x) \in -\mathbf{K}] \\ x & \in & X; \end{array} \\
\text{(ConII)}: & \\
& \begin{array}{rcl} \inf\limits_{x \in X} \left[f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x) \right] & \geq & 0 \\ \overline{\lambda} & \geq & 0, \\ \widehat{\lambda} & \geq_{\mathbf{K}_*} & 0 \ [\Leftrightarrow \widehat{\lambda} \in \mathbf{K}_*] \end{array}
\end{array}
$$

We are in the case when $\overline{g}(\bar{x}) \leq 0$ and $\widehat{g}(\bar{x}) \in -\operatorname{int} \mathbf{K}$ for some $\bar{x} \in \operatorname{rint} X = \operatorname{int} X$; by shifting X (this clearly does not affect the statement we need to prove) we can assume that $\bar{x}$ is just the origin, so that

$$
0 \in \operatorname{int} X, \ b \geq 0, \ \widehat{g}(0) <_{\mathbf{K}} 0. \tag{3.2.6}
$$

Recall that we are in the situation when (ConI) is infeasible, that is, the optimization problem

$$
\operatorname{Opt}(P) = \min \{ f(x) : x \in X, \overline{g}(x) \leq 0, \widehat{g}(x) \leq_{\mathbf{K}} 0 \} \tag{P}
$$

satisfies $\operatorname{Opt}(P) \geq 0$, and our goal is to show that (ConII) is feasible.

1^0. Let $Y = \{x \in X : \overline{g}(x) \leq 0\} = \{x \in X : Ax - b \leq 0\}$,

$$
\begin{array}{rcl}
S & = & \{t = [t_0; t_1] \in \mathbf{R} \times \mathbf{R}^\nu : t_0 < 0, t_1 \leq_{\mathbf{K}} 0\}, \\
T & = & \{t = [t_0; t_1] \in \mathbf{R} \times \mathbf{R}^\nu : \exists x \in Y : f(x) \leq t_0, \ \widehat{g}(x) \leq_{\mathbf{K}} t_1\},
\end{array}
$$

so that S and T are nonempty (since (P) is feasible by (3.2.6)) non-intersecting (since $\operatorname{Opt}(P) \geq 0$) convex (since X and f are convex, and $\widehat{g}(x)$ is $\mathbf{K}$-convex on X) sets. By Separation Theorem (Theorem 1.2.9) S and T can be separated by an appropriately chosen linear form α. Thus,

$$
\sup_{t \in S} \alpha^T t \leq \inf_{t \in T} \alpha^T t \tag{3.2.7}
$$

and the form is non-constant on $S \cup T$, implying that $\alpha \neq 0$. We clearly have $\alpha \in \mathbf{R}_+ \times \mathbf{K}_*$, since otherwise $\sup_{t \in S} \alpha^T t$ would be $+\infty$. Since $\alpha \in \mathbf{R}_+ \times \mathbf{K}_*$, the left hand side in (3.2.7) is 0 (look what S is), and therefore (3.2.7) reads

$$
0 \leq \inf_{t \in T} \alpha^T t. \tag{3.2.8}
$$

2^0. Denoting $\alpha = [\alpha_0; \alpha_1]$ with $\alpha_0 \in \mathbf{R}_+$ and $\alpha_1 \in \mathbf{K}_*$, we claim that $\alpha_0 > 0$. Indeed, the point $\bar{t} = [\bar{t}_0; \bar{t}_1]$ with the components $\bar{t}_0 = f(0)$ and $\bar{t}_1 = \widehat{g}(0)$ belongs to T, so that by (3.2.8) it holds

$\alpha_0\bar{t}_0 + \alpha_1^T\bar{t}_1 \geq 0$. Assuming that $\alpha_0 = 0$, it follows that $\alpha_1^T\bar{t}_1 \geq 0$, and since $\bar{t}_1 \in -\operatorname{int}\mathbf{K}$ (see (3.2.6)) and $\alpha_1 \in \mathbf{K}_*$, we conclude (see observation (!) on p. 47) that $\alpha_1 = 0$ on the top of $\alpha_0 = 0$, which is impossible, since $\alpha \neq 0$.

In the sequel, we set $\bar{\alpha}_1 = \alpha_1/\alpha_0$, and

$$h(x) = f(x) + \bar{\alpha}_1^T\widehat{g}(x).$$

Observing that (3.2.8) remains valid when replacing α with $\bar{\alpha} = \alpha/\alpha_0$ and that the vector $[f(x); \widehat{g}(x)]$ belongs to T when $x \in Y$, we conclude that

$$x \in Y \Rightarrow h(x) \geq 0. \tag{3.2.9}$$

3^0.1. Consider the convex sets

$$S = \{[x;\tau] \in \mathbf{R}^n \times \mathbf{R} : x \in X, \overline{g}(x) := Ax - b \leq 0, \tau < 0\},\ T = \{[x;\tau] \in \mathbf{R}^n \times \mathbf{R} : x \in X, \tau \geq h(x)\}.$$

These sets clearly are nonempty and do not intersect (since the x-component x of a point from $S \cap T$ would satisfy the premise and violate the conclusion in (3.2.9)). By Separation Theorem, there exists $[e;\alpha] \neq 0$ such that

$$\sup_{[x;\tau]\in S}[e^Tx + \alpha\tau] \leq \inf_{[x;\tau]\in T}[e^Tx + \alpha\tau].$$

Taking into account what S is, we have $\alpha \geq 0$ (since otherwise the left hand side in the inequality would be $+\infty$). With this in mind, the inequality reads

$$\sup_x\left\{e^Tx : x \in X, \overline{g}(x) \leq 0\right\} \leq \inf_x\left\{e^Tx + \alpha h(x) : x \in X\right\}. \tag{3.2.10}$$

3^0.2. We need the following fact:

Lemma 3.2.1 *Let $X \subset \mathbf{R}^n$ be a convex set with $0 \in \operatorname{int} X$, $g(x) = Ax + a$ be such that $g(0) \leq 0$, and $d^Tx + \delta$ be an affine function such that*

$$x \in X, g(x) \leq 0 \Rightarrow d^Tx + \delta \leq 0.$$

Then there exists $\mu \geq 0$ such that

$$d^Tx + \delta \leq \mu^Tg(x)\ \forall x \in X.$$

Note that when $X = \mathbf{R}^n$, Lemma 3.2.1 is nothing but Inhomogeneous Farkas Lemma.

Proof of Lemma 3.2.1. Consider the cones

$$M_1 = \operatorname{cl}\{[x;t] : t > 0, x/t \in X\},\ M_2 = \{y = [x;t] : Ax + ta \leq 0\}.$$

M_2 is a polyhedral cone, and M_1 is a closed convex cone with a nonempty interior (since the point $[0;1]$ belongs to $\operatorname{int} M_1$ due to $0 \in \operatorname{int} X$); moreover, $[\operatorname{int} M_1] \cap M_2 \neq \emptyset$ (since the point $[0;1] \in \operatorname{int} M_1$ belongs to M_2 due to $g(0) \leq 0$). Observe that the linear form $f^T[x;t] := d^Tx + t\delta$ is nonpositive on $M_1 \cap M_2$, that is, $(-f) \in [M_1 \cap M_2]_*$, where, as usual, M_* is the cone dual to a cone M.

Indeed, let $[z;t] \in M_1 \cap M_2$, and let $y_s = [z_s;t_s] = (1-s)y + s[0;1]$. Observe that $y_s \in M_1 \cap M_2$ when $0 \leq s \leq 1$. When $0 < s \leq 1$, we have $t_s > 0$, and $y_s \in M_1$ implies that $w_s := z_s/t_s \in \operatorname{cl} X$ (why?), while $y_s \in M_2$ implies that $g(w_s) \leq 0$. Since $0 \in \operatorname{int} X$, $w_s \in \operatorname{cl} X$ implies that $\theta w_s \in X$ for all $\theta \in [0,1)$, while $g(0) \leq 0$ along with $g(w_s) \leq 0$ implies that $g(\theta w_s) \leq 0$ for $\theta \in [0,1)$. Invoking the premise of Lemma, we conclude that

$$d^T(\theta w_s) + \delta \leq 0\ \forall \theta \in [0,1],$$

whence $d^Tw_s + \delta \leq 0$, or, which is the same, $f^Ty_s \leq 0$. As $s \to +0$, we have $f^Ty_s \to f^T[z;t]$, implying that $f^T[z;t] \leq 0$, as claimed.

Thus, M_1, M_2 are closed cones such that $[\text{int } M_1] \cap M_2 \neq \emptyset$ and $f \in [M_1 \cap M_2]*$. Applying to M_1, M_2 the Dubovitski-Milutin Lemma (Proposition 1.2.7), we conclude that $[M_1 \cap M_2]_* = (M_1)_* + (M_2)_*$. Since $-f \in [M_1 \cap M_2]*$, there exist $\psi \in (M_1)_*$ and $\phi \in (M_2)_*$ such that $[d; \delta] = -\phi - \psi$. The inclusion $\phi \in (M_2)_*$ means that the homogeneous linear inequality $\phi^T y \geq 0$ is a consequence of the system $[A, a]y \leq 0$ of homogeneous linear inequalities; by Homogeneous Farkas Lemma (Lemma 1.2.1), it follows that identically in x it holds $\phi^T[x; 1] = -\mu^T g(x)$, with some nonnegative μ. Thus,

$$\forall x \in E : d^T x + \delta = [d; \delta]^T [x; 1] = \mu^T g(x) - \psi^T [x; 1],$$

and since $[x; 1] \in M_1$ whenever $x \in X$, we have $\psi^T[x; 1] \geq 0$, $x \in X$, so that μ satisfies the requirements stated in the lemma we are proving. ∎

3^0.3. Recall that we have seen that in (3.2.10), $\alpha \geq 0$. We claim that in fact $\alpha > 0$.

Indeed, assuming $\alpha = 0$ and setting $a = \sup_x \left\{e^T x : x \in X, \overline{g}(x) \leq 0\right\}$, (3.2.10) implies that $a \in \mathbf{R}$ and

$$\{\forall (x \in X, \overline{g}(x) \leq 0) : e^T x - a \leq 0\} \ \& \ \{\forall x \in X : e^T x \geq a\}. \tag{3.2.11}$$

By Lemma 3.2.1, the first of these relations implies that for some nonnegative μ it holds

$$e^T x - a \leq \mu^T \overline{g}(x) \ \forall x \in X.$$

Setting $x = 0$, we get $a \geq 0$ due to $\mu \geq 0$ and $\overline{g}(0) \leq 0$, so that the second relation in (3.2.11) implies $e^T x \geq 0$ for all $x \in X$, which is impossible: since $0 \in \text{int } X$, $e^T x \geq 0$ for all $x \in X$ would imply that $e = 0$, and since we are in the case of $\alpha = 0$, we would get $[e; \alpha] = 0$, which is not the case due to the origin of $[e; \alpha]$.

3^0.4. Thus, α in (3.2.10) is strictly positive; replacing e with $\alpha^{-1}e$, we can assume that (3.2.10) holds true with $\alpha = 1$. Setting $a = \sup_x \left\{e^T x : x \in X, \overline{g}(x) \leq 0\right\}$, (3.2.10) reads

$$\forall (x \in X) : h(x) + e^T x \geq a,$$

while the definition of a along with Lemma 3.2.1 implies that there exists $\mu \geq 0$ such that

$$\forall x \in X : e^T x - a \leq \mu^T \overline{g}(x).$$

Combining these relations, we conclude that

$$h(x) + \mu^T \overline{g}(x) \geq 0 \ \forall x \in X. \tag{3.2.12}$$

Recalling that $h(x) = f(x) + \bar{\alpha}_1^T \widehat{g}(x)$ with $\bar{\alpha}_1 \in \mathbf{K}_*$ and setting $\overline{\lambda} = \mu$, $\widehat{\lambda} = \bar{\alpha}_1$, we get $\overline{\lambda} \geq 0$, $\widehat{\lambda} \in \mathbf{K}_*$ while by (3.2.12) it holds

$$f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x) \geq 0 \ \forall x \in X,$$

meaning that $\overline{\lambda}, \widehat{\lambda}$ solve (ConII) (recall that we are in the case of $c = 0$). ∎

3.2.2 Lagrange Function and Lagrange Duality

3.2.2.1 Lagrange function

Convex Theorem on Alternative brings to our attention the function

$$\underline{L}(\lambda) = \inf_{x \in X} \left[f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) \right], \tag{3.2.13}$$

same as the aggregate

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) \tag{3.2.14}$$

from which this function comes. Aggregate (3.2.14) is called the *Lagrange function* of the inequality constrained optimization program

$$\text{(IC)} \qquad \min\{f(x) : g_j(x) \le 0,\ j = 1, ..., m,\ x \in X\}.$$

The Lagrange function of an optimization program is a very important entity: most of optimality conditions are expressed in terms of this function. Let us start with translating of what we already know to the language of the Lagrange function.

3.2.2.2 Convex Programming Duality Theorem

Theorem 3.2.3 *Consider an arbitrary inequality constrained optimization program* (IC). *Then*

(i) *The infimum*

$$\underline{L}(\lambda) = \inf_{x \in X} L(x, \lambda)$$

of the Lagrange function in $x \in X$ is, for every $\lambda \ge 0$, a lower bound on the optimal value in (IC), *so that the optimal value in the optimization program*

$$\text{(IC}^*\text{)} \qquad \sup_{\lambda \ge 0} \underline{L}(\lambda)$$

also is a lower bound for the optimal value in (IC);

(ii) [Convex Duality Theorem] *If* (IC)

- *is convex,*
- *is below bounded*

and

- *satisfies the Relaxed Slater condition,*

then the optimal value in (IC*) *is attained and is equal to the optimal value in* (IC).

Proof. (i) is nothing but Proposition 3.2.1 (why?). It makes sense, however, to repeat here the corresponding one-line reasoning:

> Let $\lambda \ge 0$; in order to prove that
>
> $$\underline{L}(\lambda) \equiv \inf_{x \in X} L(x, \lambda) \le c^* \quad [L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x)],$$
>
> where c^* is the optimal value in (IC), note that if x is feasible for (IC), then evidently $L(x, \lambda) \le f(x)$, so that the infimum of L over $x \in X$ is $\le$ the infimum c^* of f over the feasible set of (IC). □

(ii) is an immediate consequence of the Convex Theorem on Alternative. Indeed, let c^* be the optimal value in (IC). Then the system

$$f(x) < c^*, g_j(x) \le 0,\ j = 1, ..., m$$

has no solutions in X, and by the above Theorem the system (II) associated with $c = c^*$ has a solution, i.e., there exists $\lambda^* \ge 0$ such that $\underline{L}(\lambda^*) \ge c^*$. But we know from (i) that the strict inequality here is impossible and, besides this, that $\underline{L}(\lambda) \le c^*$ for every $\lambda \ge 0$. Thus, $\underline{L}(\lambda^*) = c^*$ and λ^* is a maximizer of $\underline{L}$ over $\lambda \ge 0$. ■

3.2.2.3 Dual program

Theorem 3.2.3 establishes certain connection between two optimization programs – the "primal" program

$$\text{(IC)} \qquad \min\{f(x) : g_j(x) \le 0,\ j = 1, ..., m,\ x \in X\}$$

and its *Lagrange dual program*

$$\text{(IC}^*\text{)} \qquad \max\left\{\underline{L}(\lambda) \equiv \inf_{x\in X} L(x,\lambda) : \lambda \ge 0\right\}$$

(the variables λ of the dual problem are called the *Lagrange multipliers* of the primal problem). The Theorem says that the optimal value in the dual problem is $\le$ the one in the primal, and under some favourable circumstances (the primal problem is convex below bounded and satisfies the Slater condition) the optimal values in the programs are equal to each other.

In our formulation there is some asymmetry between the primal and the dual programs. In fact both of the programs are related to the Lagrange function in a quite symmetric way. Indeed, consider the program

$$\min_{x\in X} \overline{L}(x), \quad \overline{L}(x) = \sup_{\lambda\ge 0} L(\lambda, x).$$

The objective in this program clearly is $+\infty$ at every point $x \in X$ which is not feasible for (IC) and is $f(x)$ on the feasible set of (IC), so that the program is equivalent to (IC). We see that both the primal and the dual programs come from the Lagrange function: in the primal problem, we minimize over X the result of maximization of $L(x,\lambda)$ in $\lambda \ge 0$, and in the dual program we maximize over $\lambda \ge 0$ the result of minimization of $L(x,\lambda)$ in $x \in X$. This is a particular (and the most important) example of a *zero sum two person game* – the issue we will speak about later.

We have seen that under certain convexity and regularity assumptions the optimal values in (IC) and (IC*) are equal to each. There is also another way to say when these optimal values are equal – this is always the case when the Lagrange function possesses a saddle point, i.e., there exists a pair $x^* \in X, \lambda^* \ge 0$ such that at the pair $L(x,\lambda)$ attains its minimum as a function of $x \in X$ and attains its maximum as a function of $\lambda \ge 0$:

$$L(x,\lambda^*) \ge L(x^*,\lambda^*) \ge L(x^*,\lambda) \quad \forall x \in X, \lambda \ge 0.$$

It can be easily demonstrated (do it by yourself or look at Theorem 3.4.1) that

Proposition 3.2.2 *(x^*,λ^*) is a saddle point of the Lagrange function L of* (IC) *if and only if x^* is an optimal solution to* (IC)*, λ^* is an optimal solution to* (IC*) *and the optimal values in the indicated problems are equal to each other.*

3.2.2.4 Cone constrained forms of Lagrange Function, Lagrange Duality, and Convex Programming Duality Theorem

The above results related to convex optimization problems in the standard MP format admit instructive extensions to the case of convex problems in cone constrained form. These extensions are as follows:

3.2.2.4.1. Convex problem in cone constrained form is optimization problem of the form

$$\mathrm{Opt}(P) = \min_{x\in X}\{f(x) : \overline{g}(x) := Ax - b \le 0, \widehat{g}(x) \le_{\mathbf{K}} 0\} \qquad (P)$$

where $X \subset \mathbf{R}^n$ is a nonempty convex set, $f : X \to \mathbf{R}$ is a convex function, $\mathbf{K} \subset \mathbf{R}^\nu$ is a regular cone, and $\widehat{g}(\cdot) : X \to \mathbf{R}^\nu$ is $\mathbf{K}$-convex.

3.2.2.4.2. Cone constrained Lagrange function of (P) is

$$L(x;\lambda:=[\overline{\lambda};\widehat{\lambda}])=f(x)+\overline{\lambda}^T\overline{g}(x)+\widehat{\lambda}^T\widehat{g}(x):X\times\overbrace{[\mathbf{R}^k_+\times\mathbf{K}_*]}^{\Lambda}\to\mathbf{R}\qquad[k=\dim b]$$

By construction, for $\lambda\in\Lambda$ $L(x;\lambda)$ as a function of x underestimates $f(x)$ everywhere on X.

3.2.2.4.3. Cone constrained Lagrange dual of (P) is the optimization problem

$$\mathrm{Opt}(D)=\max\left\{\underline{L}(\lambda):=\inf_{x\in X}L(x;\lambda):\lambda\in\Lambda:=\mathbf{R}^k_+\times\mathbf{K}_*\right\}\qquad(D)$$

From $L(x;\lambda)\le f(x)$ for all $x\in X,\lambda\in\Lambda$ we clearly extract that

$$\mathrm{Opt}(D)\le\mathrm{Opt}(P);\qquad\text{[Weak Duality]}$$

note that this fact is independent of any assumptions of convexity on f, F and on $\mathbf{K}$-convexity of $\widehat{g}$.

3.2.2.4.4. Convex Programming Duality Theorem in cone constrained form reads:

Theorem 3.2.4 [Convex Programming Duality Theorem in cone constrained form] *Consider convex cone constrained problem (P), that is, X is convex, f is real-valued and convex on X, and $\widehat{g}(\cdot)$ is well defined and $\mathbf{K}$-convex on X. Assume that the problem is below bounded and satisfies the Relaxed Slater condition. Then (D) is solvable and $\mathrm{Opt}(P)=\mathrm{Opt}(D)$.*

Note that the only nontrivial part (ii) of Theorem 3.2.3 is nothing but the special case of Theorem 3.2.4 where $\mathbf{K}$ is a nonnegative orthant.

Proof of Theorem 3.2.4 is immediate. Under the premise of Theorem, $c:=\mathrm{Opt}(P)$ is a real, and the associated with this c system of constraints (ConI) has no solutions. Relaxed Slater Condition combines with Convex Theorem on Alternative in cone constrained form (Theorem 3.2.2) to imply solvability of (ConII), or, which is the same, existence of $\lambda_*=[\overline{\lambda}_*;\widehat{\lambda}_*]\in\Lambda$ such that

$$\underline{L}(\lambda_*)=\inf_{x\in X}\left\{f(x)+\overline{\lambda}_*^T\overline{g}(x)+\widehat{\lambda}_*^T\widehat{g}(x)\right\}\ge c=\mathrm{Opt}(P).$$

We see that (D) has a feasible solution with the value of the objective $\ge\mathrm{Opt}(P)$, By Weak Duality, this value is exactly $\mathrm{Opt}(P)$, the solution in question is optimal for (D), and $\mathrm{Opt}(P)=\mathrm{Opt}(D)$. ∎

3.2.2.5 Conic Programming and Conic Duality Theorem

Consider the special case of convex problem in cone constrained form – one where $f(x)$ is linear, X is the entire space and $\widehat{g}(x)=Px-p$ is affine; in today optimization, problems of this type are called *conic*. Note that affine mapping is $\mathbf{K}$-convex whatever be the cone $\mathbf{K}$. Thus, conic problem automatically satisfies convexity restrictions from Conic Lagrange Duality Theorem and is optimization problem of the form

$$\mathrm{Opt}(P)=\min_{x\in\mathbf{R}^n}\left\{c^Tx:Ax-b\le0,\ Px-p\le_{\mathbf{K}}0\right\}$$

where $\mathbf{K}$ is a regular cone in certain $\mathbf{R}^\nu$. The Conic Lagrange dual problem reads

$$\max_{\overline{\lambda},\widehat{\lambda}}\left[\inf_x\left\{c^Tx+\overline{\lambda}^T[Ax-b]+\widehat{\lambda}^T[Px-p]:\overline{\lambda}\ge0,\widehat{\lambda}\in\mathbf{K}_*\right\}\right],$$

which, as it is immediately seen, is nothing but the problem

$$\mathrm{Opt}(D)=\max_{\overline{\lambda},\widehat{\lambda}}\left\{-b^T\overline{\lambda}-p^T\widehat{\lambda}:A^T\overline{\lambda}+P^T\widehat{\lambda}+c=0,\ \overline{\lambda}\ge0,\widehat{\lambda}\in\mathbf{K}_*\right\}\qquad(D)$$

It is easy to show that the cone dual to a regular cone also is a regular cone; as a result, problem (D), called the *conic dual* of conic problem (P) indeed is a conic problem. An immediate computation (utilizing the fact that $[\mathbf{K}_*]_*=\mathbf{K}$ for every regular cone $\mathbf{K}$) shows that *conic duality if symmetric* – the conic dual to (D) is (equivalent to) (P).

Indeed, rewriting (D) in the minimization form with ≤ 0 polyhedral constraints, as required by our recipe for building the conic dual to a conic problem, we get

$$-\mathrm{Opt}(D) = \min_{\overline{\lambda},\widehat{\lambda}} \left\{ b^T\overline{\lambda} + p^T\widehat{\lambda} : \left\{ \begin{array}{l} A^T\overline{\lambda} + P^T\widehat{\lambda} + c \le 0 \\ -A^T\overline{\lambda} - P^T\widehat{\lambda} - c \le 0 \\ -\overline{\lambda} \le 0 \end{array} , -\widehat{\lambda} \in -\mathbf{K}_* \right\} \right. \qquad (D)$$

The dual to this problem, in view of $[\mathbf{K}_*]_* = \mathbf{K}$, reads

$$\max_{u,v,w,y} \left\{ c^T[u-v] : b + A[u-v] - w = 0, P[u-v] + p - y = 0, u \ge 0, v \ge 0, w \ge 0, y \in \mathbf{K} \right\}.$$

After setting $x = v - u$ and eliminating y and w, the latter problem becomes

$$\max_x \left\{ -c^T x : Ax - b \le 0, Px - p \le_{\mathbf{K}} 0 \right\},$$

which is noting but (P).

In view of primal-dual symmetry, Convex Conic Duality Theorem in the case in question takes the following nice form:

Theorem 3.2.5 [Conic Duality Theorem] *Consider a primal-dual pair of conic problems*

$$\begin{array}{rcll} \mathrm{Opt}(P) & = & \min_{x \in \mathbf{R}^n} \left\{ c^T x : Ax - b \le 0,\ Px - x \le_{\mathbf{K}} 0 \right\} & (P) \\ \mathrm{Opt}(D) & = & \max_{\overline{\lambda},\widehat{\lambda}} \left\{ -b^T\overline{\lambda} - p^T\widehat{\lambda} : A^T\overline{\lambda} + P^T\widehat{\lambda} + c = 0,\ \overline{\lambda} \ge 0, \widehat{\lambda} \in \mathbf{K}_* \right\} & (D) \end{array}$$

One always has $\mathrm{Opt}(D) \le \mathrm{Opt}(P)$. *Besides this, if one of the problems in the pair is bounded and satisfies the Relaxed Slater condition, then the other problem in the pair is solvable, and* $\mathrm{Opt}(P) = \mathrm{Opt}(D)$. *Finally, if both the problems satisfy Relaxed Slater condition, then both are solvable with equal optimal values.*

Proof is immediate. Weak duality has already been verified. To verify the second claim, note that by primal-dual symmetry we can assume that the bounded problem satisfying Relaxed Slater condition is (P); but then the claim in question is given by Theorem 3.2.4. Finally, if both problems satisfy Relaxed Slater condition (and in particular are feasible), both are bounded, and therefore solvable with equal optimal values by the second claim. ∎

3.2.3 Optimality Conditions in Convex Programming

Our current goal is to extract from what we already know optimality conditions for convex programs.

3.2.3.1 Saddle point form of optimality conditions

Theorem 3.2.6 [Saddle Point formulation of Optimality Conditions in Convex Programming] *Let* (IC) *be an optimization program,* $L(x,\lambda)$ *be its Lagrange function, and let* $x^* \in X$. *Then*

(i) *A sufficient condition for* x^* *to be an optimal solution to* (IC) *is the existence of the vector of Lagrange multipliers* $\lambda^* \ge 0$ *such that* (x^*,λ^*) *is a saddle point of the Lagrange function* $L(x,\lambda)$, *i.e., a point where* $L(x,\lambda)$ *attains its minimum as a function of* $x \in X$ *and attains its maximum as a function of* $\lambda \ge 0$*:*

$$L(x,\lambda^*) \ge L(x^*,\lambda^*) \ge L(x^*,\lambda) \quad \forall x \in X, \lambda \ge 0. \qquad (3.2.15)$$

(ii) *if the problem* (IC) *is convex and satisfies the Relaxed Slater condition, then the above condition is necessary for optimality of* x^**: if* x^* *is optimal for* (IC), *then there exists* $\lambda^* \ge 0$ *such that* (x^*,λ^*) *is a saddle point of the Lagrange function.*

Proof. (i): assume that for a given $x^* \in X$ there exists $\lambda^* \geq 0$ such that (3.2.15) is satisfied, and let us prove that then x^* is optimal for (IC). First of all, x^* is feasible: indeed, if $g_j(x^*) > 0$ for some j, then, of course, $\sup_{\lambda \geq 0} L(x^*, \lambda) = +\infty$ (look what happens when all λ's, except λ_j, are fixed, and $\lambda_j \to +\infty$); but $\sup_{\lambda \geq 0} L(x^*, \lambda) = +\infty$ is forbidden by the second inequality in (3.2.15).

Since x^* is feasible, $\sup_{\lambda \geq 0} L(x^*, \lambda) = f(x^*)$, and we conclude from the second inequality in (3.2.15) that $L(x^*, \lambda^*) = f(x^*)$. Now the first inequality in (3.2.15) reads

$$f(x) + \sum_{j=1}^{m} \lambda_j^* g_j(x) \geq f(x^*) \quad \forall x \in X.$$

This inequality immediately implies that x^* is optimal: indeed, if x is feasible for (IC), then the left hand side in the latter inequality is $\leq f(x)$ (recall that $\lambda^* \geq 0$), and the inequality implies that $f(x) \geq f(x^*)$. □

(ii): Assume that (IC) is a convex program, x^* is its optimal solution and the problem satisfies the Relaxed Slater condition; we should prove that then there exists $\lambda^* \geq 0$ such that (x^*, λ^*) is a saddle point of the Lagrange function, i.e., that (3.2.15) is satisfied. As we know from the Convex Programming Duality Theorem (Theorem 3.2.3.(ii)), the dual problem (IC*) has a solution $\lambda^* \geq 0$ and the optimal value of the dual problem is equal to the optimal value in the primal one, i.e., to $f(x^*)$:

$$f(x^*) = \underline{L}(\lambda^*) \equiv \inf_{x \in X} L(x, \lambda^*). \tag{3.2.16}$$

We immediately conclude that

$$\lambda_j^* > 0 \Rightarrow g_j(x^*) = 0$$

(this is called *complementary slackness*: positive Lagrange multipliers can be associated only with active (satisfied at x^* as equalities) constraints). Indeed, from (3.2.16) it for sure follows that

$$f(x^*) \leq L(x^*, \lambda^*) = f(x^*) + \sum_{j=1}^{m} \lambda_j^* g_j(x^*);$$

the terms in the $\sum_j$ in the right hand side are nonpositive (since x^* is feasible for (IC)), and the sum itself is nonnegative due to our inequality; it is possible if and only if all the terms in the sum are zero, and this is exactly the complementary slackness.

From the complementary slackness we immediately conclude that $f(x^*) = L(x^*, \lambda^*)$, so that (3.2.16) results in

$$L(x^*, \lambda^*) = f(x^*) = \inf_{x \in X} L(x, \lambda^*).$$

On the other hand, since x^* is feasible for (IC), we have $L(x^*, \lambda) \leq f(x^*)$ whenever $\lambda \geq 0$. Combining our observations, we conclude that

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*)$$

for all $x \in X$ and all $\lambda \geq 0$. ■

Note that (i) is valid for an arbitrary inequality constrained optimization program, not necessarily convex. However, in the nonconvex case the *sufficient* condition for optimality given by (i) is extremely far from being necessary and is "almost never" satisfied. In contrast to this, in the convex case the condition in question is not only sufficient, but also "nearly necessary" – it for sure is necessary when (IC) is a convex program satisfying the Relaxed Slater condition

We have established Saddle Point form of optimality conditions for convex problems in the standard Mathematical Programming form (that is, with constraints represented by scalar convex inequalities). Similar results can be obtained for convex cone constrained problems:

Theorem 3.2.7 [Saddle Point formulation of Optimality Conditions in Convex Conic Programming] *Consider a convex cone constrained problem*

$$\mathrm{Opt}(P) = \min_{x\in X}\{f(x) : \overline{g}(x) := Ax - b \le 0, \widehat{g}(x) \le_{\mathbf{K}} 0\} \qquad (P)$$

(X is convex, $f : X \to \mathbf{R}$ is convex, $\mathbf{K}$ is a regular cone, and $\widehat{g}$ is $\mathbf{K}$-convex on X) along with its Conic Lagrange dual problem

$$\mathrm{Opt}(D) = \max_{\lambda=[\overline{\lambda};\widehat{\lambda}]} \left\{ \overline{L}(\lambda) := \inf_{x\in X} \left[f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x) \right] : \overline{\lambda} \ge 0, \widehat{\lambda} \in \mathbf{K}_* \right\} \qquad (D)$$

and assume that (P) is bounded and satisfies Relaxed Slater condition. Then a point $x_* \in X$ is an optimal solution to (P) is and only if x_* can be augmented by properly selected $\lambda \in \Lambda := \mathbf{R}_+ \times [\mathbf{K}_*]$ to a saddle pint of the cone constrained Lagrange function

$$L(x; \lambda := [\overline{\lambda}; \widehat{\lambda}]) := f(x) + \overline{\lambda}^T \overline{g}(x) + \widehat{\lambda}^T \widehat{g}(x)$$

on $X \times \Lambda$.

Proof repeats with evident modifications the one of relevant part of Theorem 3.2.6, with Conic Programming Duality Theorem (Theorem 3.2.4) in the role of Convex Programming Duality Theorem (Theorem 3.2.3). ∎

3.2.3.2 Karush-Kuhn-Tucker form of optimality conditions

Theorem 3.2.6 provides, basically, the strongest optimality conditions for a Convex Programming program. These conditions, however, are "implicit" – they are expressed in terms of saddle point of the Lagrange function, and it is unclear how to verify that something is or is not the saddle point of the Lagrange function. Fortunately, the proof of Theorem 3.2.6 yields more or less explicit optimality conditions. We start with a useful

Definition 3.2.3 [Normal Cone] *Let $X \subset \mathbf{R}^n$ and $x \in X$. The normal cone of $N_X(x)$ of X taken at the point x is the set*

$$N_X(x) = \{h \in \mathbf{R}^n : h^T(x' - x) \ge 0 \, \forall x' \in X\}.$$

Note that $N_X(x)$ indeed is a closed convex cone.
Examples:

1. when $x \in \mathrm{int}\, X$, we have $N_X(x) = \{0\}$;
2. when $x \in \mathrm{rint}\, X$, we have $N_X(x) = L^\perp$, where $L = \mathrm{Lin}(X - x)$ is the linear subspace parallel to the affine hull $\mathrm{Aff}(X)$ of X.

Theorem 3.2.8 [Karush-Kuhn-Tucker Optimality Conditions in Convex Programming] *Let* (IC) *be a convex program, let x^* be its feasible solution, and let the functions f, g_1,...,g_m be differentiable at x^*. Then*

(i) [Sufficiency] *The Karush-Kuhn-Tucker condition:*

There exist nonnegative Lagrange multipliers λ_j^*, $j = 1, ..., m$, such that

$$\lambda_j^* g_j(x^*) = 0, \; j = 1, ..., m \quad \text{[complementary slackness]} \qquad (3.2.17)$$

and

$$\nabla f(x^*) + \sum_{j=1}^m \lambda_j^* \nabla g_j(x^*) \in N_X(x^*) \qquad (3.2.18)$$

(see Definition 3.2.3)

is sufficient for x^* to be optimal solution to (IC).

(ii) [Necessity and sufficiency] *If, in addition to the premise, the Relaxed Slater condition holds, then the Karush-Kuhn-Tucker condition from* (i) *is necessary and sufficient for x^* to be optimal solution to* (IC).

Proof. Observe that under the premise of Theorem 3.2.8 the Karush-Kuhn-Tucker condition is necessary and sufficient for (x^*, λ^*) to be a saddle point of the Lagrange function. Indeed, the (x^*, λ^*) is a saddle point of the Lagrange function if and only if a) $L(x^*, \lambda) = f(x^*) + \sum_{i=1}^m \lambda_j g_j(x^*)$ as a function of $\lambda \geq 0$ attains its minimum at $\lambda = \lambda^*$. Since $L(x^*, \lambda)$ is linear in λ, this is exactly the same as to say that for all j we have $g_j(x_*) \geq 0$ (which we knew in advance – x^* is feasible!) and $\lambda_j^* g_j(x^*) = 0$, which is complementary slackness;
b) $L(x, \lambda^*)$ as a function of $x \in X$ attains it minimum at x_*. Since $L(x, \lambda^*)$ is convex in $x \in X$ due to $\lambda^* \geq 0$ and $L(x, \lambda^*)$ is differentiable at x_* by Theorem's premise, Proposition 2.5.1 says that $L(x, \lambda^*)$ achieves its minimum at x^* if and only if $\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*)$ has nonnegative inner products with all vectors h from the radial cone $T_X(x^*)$, i.e., all h such that $x^* + th \in X$ for all small enough $t > 0$, which is exactly the same as to say that $\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) \in N_X(x^*)$ (since for a convex set X and all $x \in X$ it clearly holds $N_X(x) = \{f : f^T h \geq 0 \, \forall h \in T_X(x)\}$). The bottom line is that for a feasible x^* and $\lambda^* \geq 0$, (x^*, λ^*) is a saddle point of the Lagrange function if and only if (x^*, λ^*) meet the Karush-Kuhn-Tucker condition. This observation proves (i), and combined with Theorem 3.2.3, proves (ii) as well. ■

Note that the optimality conditions stated in Theorem 2.5.2 and Proposition 2.5.1 are particular cases of the above Theorem corresponding to $m = 0$.

Note that in the case when $x^* \in \operatorname{int} X$, we have $N_X(x^*) = \{0\}$, so that (3.2.18) reads

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0;$$

when $x^* \in \operatorname{rint} X$, $N_X(x^*)$ is the orthogonal complement to the linear subspace L to which $\operatorname{Aff}(X)$ is parallel, so that (3.2.18) reads

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) \text{ is orthogonal to } L = \operatorname{Lin}(X - x_*).$$

3.2.3.3 Optimality conditions in Conic Programming

Theorem 3.2.9 [Optimality Conditions in Conic Programming] *Consider a primal-dual pair of conic problems (cf. Section 3.2.2.5)*

$$\begin{array}{rcll} \operatorname{Opt}(P) & = & \min_{x \in \mathbf{R}^n} \left\{ c^T x : Ax - b \leq 0, Px - p \leq_{\mathbf{K}} 0 \right\} & (P) \\ \operatorname{Opt}(D) & = & \max_{\lambda = [\overline{\lambda}; \widehat{\lambda}]} \left\{ -b^T \overline{\lambda} - p^T \widehat{\lambda} : \overline{\lambda} \geq 0. \widehat{\lambda} \in \mathbf{K}_*, A^T \overline{\lambda} + P^T \widehat{\lambda} + c = 0 \right\} & (D) \end{array}$$

and assume that both problems satisfy Relaxed Slater condition. Then a pair of feasible solutions x_* to (P) and $\lambda_* := [\overline{\lambda}_*; \widehat{\lambda}_*]$ to (D) is comprised of optimal solutions to the respective problems
— [Zero Duality Gap] *if and only if*

$$\operatorname{DualityGap}(x_*; \lambda_*) := c^T x + * - [-b^T \overline{\lambda}_* - p^T \widehat{\lambda}_*] = 0,$$

and
— [Complementary Slackness] *if and only if*

$$\overline{\lambda}_*^T [b - Ax_*] + \widehat{\lambda}^T [p - Px_*] = 0$$

Note: Under the premise of Theorem, from feasibility of x_* and λ_* for respective problems it follows that $b - Ax \geq 0$ and $p - Px \in \mathbf{K}$. Therefore Complementary slackness (which says that the sum of two

inner products, every one of a vector from a regular cone and a vector from the dual of this cone, and as such automatically nonnegative) is zero is a really strong restriction,

Proof of Theorem 3.2.9 is immediate. By Conic Duality Theorem (Theorem 3.2.5) we are in the case when $\mathrm{Opt}(P) = \mathrm{Opt}(D) \in \mathbf{R}$, and therefore

$$\mathrm{DualityGap}(x_*; \lambda_*) = [c^T x - \mathrm{Opt}(P)] + [\mathrm{Opt}(D] - [-b^T \overline{\lambda}_* - p^T \widehat{\lambda}_*]]$$

is the sum of nonoptimalities, in terms of objective, of x_* as a candidate solution to (P) and for λ_* as a candidate solutio to (D). since the candidate solutions in question are feasible for the respective problems, the sum of nonoptimalities is always nonnegative and is 0 if and only if the solutions in question are optimal for the respective problems.

It remains to note that Complementary Slackness condition is equivalent to Zero Duality Gap one, since for feasible for the respective problems x_* and λ_* we gave

$$\mathrm{DualityGap}(x_*; \lambda_*) = c^T x_* + b^T \overline{\lambda}_* + p^T \widehat{\lambda}_* = -[A^T \overline{\lambda}_* + P^T \widehat{\lambda^*}]^T x_* + b^T \overline{\lambda}_* + p^T \widehat{\lambda}_*$$
$$= \overline{\lambda}_*^T [b - A x_*] + \widehat{\lambda}_*^T [p - P x_*],$$

so that Complementary Slackness is, for feasible for the respective problems x_* and λ_*. exactly the same as Zero Duality Gap. ∎

3.3 Duality in Linear and Convex Quadratic Programming

The fundamental role of the Lagrange function and Lagrange Duality in Optimization is clear already from the Optimality Conditions given by Theorem 3.2.6, but this role is not restricted by this Theorem only. There are several cases when we can explicitly write down the Lagrange dual, and whenever it is the case, we get a pair of explicitly formulated and closely related to each other optimization programs – the *primal-dual pair*; analyzing the problems simultaneously, we get more information about their properties (and get a possibility to solve the problems numerically in a more efficient way) than it is possible when we restrict ourselves with only one problem of the pair. The detailed investigation of Duality in "well-structured" Convex Programming – in the cases when we can explicitly write down both the primal and the dual problems – goes beyond the scope of our course (mainly because the Lagrange duality is not the best possible approach here; the best approach is given by the *Fenchel Duality* – something similar, but not identical). There are, however, simple cases when already the Lagrange duality is quite appropriate. Let us look at two of these particular cases.

3.3.1 Linear Programming Duality

Let us start with some general observation. Note that the Karush-Kuhn-Tucker condition under the assumption of the Theorem ((IC) is convex, x^* is an interior point of X, $f, g_1, ..., g_m$ are differentiable at x^*) is exactly the condition that $(x^*, \lambda^* = (\lambda_1^*, ..., \lambda_m^*))$ is a saddle point of the Lagrange function

$$L(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x) : \tag{3.3.1}$$

(3.2.17) says that $L(x^*, \lambda)$ attains its maximum in $\lambda \geq 0$, and (3.2.18) says that $L(x, \lambda^*)$ attains its at λ^* minimum in x at $x = x^*$.

Now consider the particular case of (IC) where $X = \mathbf{R}^n$ is the entire space, the objective f is convex and everywhere differentiable and the constraints $g_1, ..., g_m$ are linear. For this case, Theorem 3.2.8 says to us that the KKT (Karush-Kuhn-Tucker) condition is necessary and sufficient for optimality of x^*; as we just have explained, this is the same as to say that the necessary and sufficient condition of optimality for x^* is that x^* along with certain $\lambda^* \geq 0$ form a saddle point of the Lagrange function. Combining these observations with Proposition 3.2.2, we get the following simple result:

Proposition 3.3.1 *Let* (IC) *be a convex program with $X = \mathbf{R}^n$, everywhere differentiable objective f and linear constraints $g_1, ..., g_m$. Then x^* is optimal solution to* (IC) *if and only if there exists $\lambda^* \geq 0$ such that (x^*, λ^*) is a saddle point of the Lagrange function (3.3.1) (regarded as a function of $x \in \mathbf{R}^n$ and $\lambda \geq 0$). In particular,* (IC) *is solvable if and only if L has saddle points, and if it is the case, then both* (IC) *and its Lagrange dual*

$$(\mathrm{IC}^*): \quad \max_{\lambda} \{\underline{L}(\lambda) : \lambda \geq 0\}$$

are solvable with equal optimal values.

Let us look what this proposition says in the Linear Programming case, i.e., when (IC) is the program

$$(P) \quad \min_x \left\{ f(x) = c^T x : g_j(x) \equiv b_j - a_j^T x \leq 0,\ j = 1, ..., m \right\}.$$

In order to get the Lagrange dual, we should form the Lagrange function

$$L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda_j g_j(x) = [c - \sum_{j=1}^m \lambda_j a_j]^T x + \sum_{j=1}^m \lambda_j b_j$$

of (IC) and to minimize it in $x \in \mathbf{R}^n$; this will give us the dual objective. In our case the minimization in x is immediate: the minimal value is $-\infty$, if $c - \sum\limits_{j=1}^m \lambda_j a_j \neq 0$, and is $\sum\limits_{j=1}^m \lambda_j b_j$ otherwise. We see that the Lagrange dual is

$$(D) \quad \max_{\lambda} \left\{ b^T \lambda : \sum_{j=1}^m \lambda_j a_j = c,\ \lambda \geq 0 \right\}.$$

The problem we get is the usual LP dual to (P), and Proposition 3.3.1 is one of the equivalent forms of the Linear Programming Duality Theorem which we already know.

3.3.2 Quadratic Programming Duality

Now consider the case when the original problem is linearly constrained convex quadratic program

$$(P) \quad \min_x \left\{ f(x) = \frac{1}{2} x^T D x + c^T x : g_j(x) \equiv b_j - a_j^T x \leq 0,\ j = 1, ..., m \right\},$$

where the objective is a strictly convex quadratic form, so that $D = D^T$ is positive definite matrix: $x^T D x > 0$ whenever $x \neq 0$. It is convenient to rewrite the constraints in the vector-matrix form

$$g(x) = b - Ax \leq 0,\ b = \begin{pmatrix} b_1 \\ ... \\ b_m \end{pmatrix},\ A = \begin{pmatrix} a_1^T \\ ... \\ a_m^T \end{pmatrix}.$$

In order to form the Lagrange dual to (P) program, we write down the Lagrange function

$$\begin{array}{rcl} L(x, \lambda) & = & f(x) + \sum\limits_{j=1}^m \lambda_j g_j(x) \\ & = & c^T x + \lambda^T (b - Ax) + \frac{1}{2} x^T D x \\ & = & \frac{1}{2} x^T D x - [A^T \lambda - c]^T x + b^T \lambda \end{array}$$

and minimize it in x. Since the function is convex and differentiable in x, the minimum, if exists, is given by the Fermat rule

$$\nabla_x L(x, \lambda) = 0,$$

which in our situation becomes

$$Dx = [A^T \lambda - c].$$

Since D is positive definite, it is nonsingular, so that the Fermat equation has a unique solution which is the minimizer of $L(\cdot,\lambda)$; this solution is

$$x = D^{-1}[A^T\lambda - c].$$

Substituting the value of x into the expression for the Lagrange function, we get the dual objective:

$$\underline{L}(\lambda) = -\frac{1}{2}[A^T\lambda - c]^T D^{-1}[A^T\lambda - c] + b^T\lambda,$$

and the dual problem is to maximize this objective over the nonnegative orthant. Usually people rewrite this dual problem equivalently by introducing additional variables

$$t = -D^{-1}[A^T\lambda - c] \quad [[A^T\lambda - c]^T D^{-1}[A^T\lambda - c] = t^T Dt];$$

with this substitution, the dual problem becomes

$$(D) \quad \max_{\lambda,t}\left\{-\frac{1}{2}t^T Dt + b^T\lambda : A^T\lambda + Dt = c,\ \lambda \geq 0\right\}.$$

We see that the dual problem also turns out to be linearly constrained convex quadratic program.

Note also that in the case in question feasible problem (P) automatically is solvable[3)]

With this observation, we get from Proposition 3.3.1 the following

Theorem 3.3.1 [Duality Theorem in Quadratic Programming]
Let (P) be feasible quadratic program with positive definite symmetric matrix D in the objective. Then both (P) and (D) are solvable, and the optimal values in the problems are equal to each other.

The pair $(x;(\lambda,t))$ of feasible solutions to the problems is comprised of the optimal solutions to them

(i) *if and only if the primal objective at x is equal to the dual objective at (λ,t)* ["zero duality gap" optimality condition]

same as

(ii) *if and only if*

$$\lambda_i(Ax-b)_i = 0,\ i = 1,...,m, \quad \text{and} \quad t = -x. \tag{3.3.2}$$

Proof. (i): we know from Proposition 3.3.1 that the optimal value in minimization problem (P) is equal to the optimal value in the maximization problem (D). It follows that the value of the primal objective at any primal feasible solution is $\geq$ the value of the dual objective at any dual feasible solution, and equality is possible if and only if these values coincide with the optimal values in the problems, as claimed in (i).

(ii): Let us compute the difference Δ between the values of the primal objective at primal feasible solution x and the dual objective at dual feasible solution (λ,t):

$$\begin{array}{rcl}\Delta &=& c^Tx + \frac{1}{2}x^TDx - [b^T\lambda - \frac{1}{2}t^TDt]\\ &=& [A^T\lambda + Dt]^Tx + \frac{1}{2}x^TDx + \frac{1}{2}t^TDt - b^T\lambda\\ && [\text{since } A^T\lambda + Dt = c]\\ &=& \lambda^T[Ax-b] + \frac{1}{2}[x+t]^TD[x+t]\end{array}$$

Since $Ax - b \geq 0$ and $\lambda \geq 0$ due to primal feasibility of x and dual feasibility of (λ,t), both terms in the resulting expression for Δ are nonnegative. Thus, $\Delta = 0$ (which, by (i). is equivalent to optimality of x for (P) and optimality of (λ,t) for (D)) if and only if $\sum_{j=1}^m \lambda_j(Ax-b)_j = 0$ and $(x+t)^TD(x+t) = 0$. The first of these equalities, due to $\lambda \geq 0$ and $Ax \geq b$, is equivalent to $\lambda_j(Ax-b)_j = 0$, $j = 1,...,m$; the second, due to positive definiteness of D, is equivalent to $x + t = 0$. ∎

[3)] since its objective, due to positive definiteness of D, goes to infinity as $|x| \to \infty$, and due to the following general fact:
Let (IC) *be a feasible program with closed domain X, continuous on X objective and constraints and such that $f(x) \to \infty$ as $x \in X$ "goes to infinity" (i.e., $|x| \to \infty$). Then* (IC) *is solvable.*
You are welcome to prove this simple statement (it is among the exercises accompanying the Lecture)

3.4 Saddle Points

3.4.1 Definition and Game Theory interpretation

When speaking about the "saddle point" formulation of optimality conditions in Convex Programming, we touched a very interesting in its own right topic of Saddle Points. This notion is related to the situation as follows. Let $X \subset \mathbf{R}^n$ and $\Lambda \in \mathbf{R}^m$ be two nonempty sets, and let

$$L(x, \lambda) : X \times \Lambda \to \mathbf{R}$$

be a real-valued function of $x \in X$ and $\lambda \in \Lambda$. We say that a point $(x^*, \lambda^*) \in X \times \Lambda$ is a *saddle point* of L on $X \times \Lambda$, if L attains in this point its maximum in $\lambda \in \Lambda$ and attains at the point its minimum in $x \in X$:

$$L(x, \lambda^*) \geq L(x^*, \lambda^*) \geq L(x^*, \lambda) \quad \forall (x, \lambda) \in X \times \Lambda. \tag{3.4.1}$$

The notion of a saddle point admits natural interpretation in *game terms*. Consider what is called a *two person zero sum game* where player I chooses $x \in X$ and player II chooses $\lambda \in \Lambda$; after the players have chosen their decisions, player I pays to player II the sum $L(x, \lambda)$. Of course, I is interested to minimize his payment, while II is interested to maximize his income. What is the natural notion of the equilibrium in such a game – what are the choices (x, λ) of the players I and II such that every one of the players is not interested to vary his choice independently on whether he knows the choice of his opponent? It is immediately seen that the equilibria are exactly the saddle points of the cost function L. Indeed, if (x^*, λ^*) is such a point, than the player I is not interested to pass from x to another choice, given that II keeps his choice λ fixed: the first inequality in (3.4.1) shows that such a choice cannot decrease the payment of I. Similarly, player II is not interested to choose something different from λ^*, given that I keeps his choice x^* – such an action cannot increase the income of II. On the other hand, if (x^*, λ^*) is not a saddle point, then either the player I can decrease his payment passing from x^* to another choice, given that II keeps his choice at λ^* – this is the case when the first inequality in (3.4.1) is violated, or similarly for the player II; thus, equilibria are exactly the saddle points.

The game interpretation of the notion of a saddle point motivates deep insight into the structure of the set of saddle points. Consider the following two situations:

(A) player I makes his choice first, and player II makes his choice already knowing the choice of I;

(B) vice versa, player II chooses first, and I makes his choice already knowing the choice of II.

In the case (A) the reasoning of I is: If I choose some x, then II of course will choose λ which maximizes, for my x, my payment $L(x, \lambda)$, so that I shall pay the sum

$$\overline{L}(x) = \sup_{\lambda \in \Lambda} L(x, \lambda);$$

Consequently, my policy should be to choose x which minimizes my *loss function* L, i.e., the one which solves the optimization problem

$$\text{(I)} \qquad \min_{x \in X} \overline{L}(x);$$

with this policy my anticipated payment will be

$$\inf_{x \in X} \overline{L}(x) = \inf_{x \in X} \sup_{\lambda \in \Lambda} L(x, \lambda).$$

In the case (B), similar reasoning of II enforces him to choose λ maximizing his *profit function*

$$\underline{L}(\lambda) = \inf_{x \in X} L(x, \lambda),$$

i.e., the one which solves the optimization problem

$$\text{(II)} \qquad \max_{\lambda \in \Lambda} \underline{L}(\lambda);$$

with this policy, the anticipated profit of II is

$$\sup_{\lambda\in\Lambda}\underline{L}(\lambda)=\sup_{\lambda\in\Lambda}\inf_{x\in X}L(x,\lambda).$$

Note that these two reasonings relate to two *different* games: the one with priority of II (when making his decision, II already knows the choice of I), and the one with similar priority of I. Therefore we should not, generally speaking, expect that the anticipated loss of I in (A) is equal to the anticipated profit of II in (B). What can be guessed is that the anticipated loss of I in (B) is *less than or equal to* the anticipated profit of II in (A), since the conditions of the game (B) are better for I than those of (A). Thus, we may guess that independently of the structure of the function $L(x,\lambda)$, there is the inequality

$$\sup_{\lambda\in\Lambda}\inf_{x\in X}L(x,\lambda)\le\inf_{x\in X}\sup_{\lambda\in\Lambda}L(x,\lambda). \tag{3.4.2}$$

This inequality indeed is true; which is seen from the following reasoning:

$$\begin{aligned}&\forall y\in X:\quad \inf_{x\in X}L(x,\lambda)\le L(y,\lambda)\Rightarrow\\ &\forall y\in X:\sup_{\lambda\in\Lambda}\inf_{x\in X}L(x,\lambda)\le\sup_{\lambda\in\Lambda}L(y,\lambda)\equiv\underline{L}(y);\end{aligned}$$

consequently, the quantity $\sup\limits_{\lambda\in\Lambda}\inf\limits_{x\in X}L(x,\lambda)$ is a lower bound for the function $\underline{L}(y)$, $y\in X$, and is therefore a lower bound for the infimum of the latter function over $y\in X$, i.e., is a lower bound for $\inf\limits_{y\in X}\sup\limits_{\lambda\in\Lambda}L(y,\lambda)$.

Now let us look what happens when the game in question has a saddle point (x^*,λ^*), so that

$$L(x,\lambda^*)\ge L(x^*,\lambda^*)\ge L(x^*,\lambda)\quad\forall(x,\lambda)\in X\times\Lambda. \tag{3.4.3}$$

We claim that if it is the case, then

(*) *x^* is an optimal solution to* (I), *λ^* is an optimal solution to* (II) *and the optimal values in these two optimization problems are equal to each other* (and are equal to the quantity $L(x^*,\lambda^*)$).

Indeed, from (3.4.3) it follows that

$$\underline{L}(\lambda^*)\ge L(x^*,\lambda^*)\ge\overline{L}(x^*),$$

whence, of course,

$$\sup_{\lambda\in\Lambda}\underline{L}(\lambda)\ge\underline{L}(\lambda^*)\ge L(x^*,\lambda^*)\ge\overline{L}(x^*)\ge\inf_{x\in X}\overline{L}(x).$$

the very first quantity in the latter chain is $\le$ the very last quantity by (3.4.2), which is possible if and only if all the inequalities in the chain are equalities, which is exactly what is said by (A) and (B).

Thus, if (x^*,λ^*) is a saddle point of L, then (*) takes place. We are about to demonstrate that the inverse also is true:

Theorem 3.4.1 [Structure of the saddle point set] *Let $L:X\times Y\to\mathbf{R}$ be a function. The set of saddle points of the function is nonempty if and only if the related optimization problems* (I) *and* (II) *are solvable and the optimal values in the problems are equal to each other. If it is the case, then the saddle points of L are exactly all pairs (x^*,λ^*) with x^* being an optimal solution to* (I) *and λ^* being an optimal solution to* (II)*, and the value of the cost function $L(\cdot,\cdot)$ at every one of these points is equal to the common optimal value in* (I) *and* (II)*.*

Proof. We already have established "half" of the theorem: if there are saddle points of L, then their components are optimal solutions to (I), respectively, (II), and the optimal values in these two problems are equal to each other and to the value of L at the saddle point in question. To complete the proof, we should demonstrate that if x^* is an optimal solution to (I), λ^* is an optimal solution to (II) and the

optimal values in the problems are equal to each other, then (x^*, λ^*) is a saddle point of L. This is immediate: we have

$$\begin{array}{rcll} L(x,\lambda^*) & \geq & \underline{L}(\lambda^*) & [\text{ definition of } \underline{L}] \\ & = & \overline{L}(x^*) & [\text{by assumption}] \\ & \geq & L(x^*,\lambda) & [\text{definition of } \overline{L}] \end{array}$$

whence

$$L(x,\lambda^*) \geq L(x^*,\lambda) \quad \forall x \in X, \lambda \in \Lambda;$$

substituting $\lambda = \lambda^*$ in the right hand side of this inequality, we get $L(x,\lambda^*) \geq L(x^*,\lambda^*)$, and substituting $x = x^*$ in the right hand side of our inequality, we get $L(x^*,\lambda^*) \geq L(x^*,\lambda)$; thus, (x^*,λ^*) indeed is a saddle point of L. ∎

3.4.2 Existence of Saddle Points

It is easily seen that a "quite respectable" cost function may have no saddle points, e.g., the function $L(x,\lambda) = (x-\lambda)^2$ on the unit square $[0,1] \times [0,1]$. Indeed, here

$$\underline{L}(x) = \sup_{\lambda \in [0,1]} (x-\lambda)^2 = \max\{x^2, (1-x)^2\},$$

$$\overline{L}(\lambda) = \inf_{x \in [0,1]} (x-\lambda)^2 = 0, \ \lambda \in [0,1],$$

so that the optimal value in (I) is $\frac{1}{4}$, and the optimal value in (II) is 0; according to Theorem 3.4.1 it means that L has no saddle points.

On the other hand, there are generic cases when L has a saddle point, e.g., when

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) : X \times \mathbf{R}^m_+ \to \mathbf{R}$$

is the Lagrange function of a solvable convex program satisfying the Slater condition. Note that in this case L is convex in x for every $\lambda \in \Lambda \equiv \mathbf{R}^m_+$ and is linear (and therefore concave) in λ for every fixed X. As we shall see in a while, these are the structural properties of L which take upon themselves the "main responsibility" for the fact that in the case in question the saddle points exist. Namely, there exists the following

Theorem 3.4.2 [Existence of saddle points of a convex-concave function (Sion-Kakutani)] *Let X and Λ be convex compact sets in $\mathbf{R}^n$ and $\mathbf{R}^m$, respectively, and let*

$$L(x,\lambda) : X \times \Lambda \to \mathbf{R}$$

be a continuous function which is convex in $x \in X$ for every fixed $\lambda \in \Lambda$ and is concave in $\lambda \in \Lambda$ for every fixed $x \in X$. Then L has saddle points on $X \times \Lambda$.

Proof. According to Theorem 3.4.1, we should prove that

- (i) Optimization problems (I) and (II) are solvable
- (ii) the optimal values in (I) and (II) are equal to each other.

(i) is valid independently of convexity-concavity of L and is given by the following routine reasoning from the Analysis:

Since X and Λ are compact sets and L is continuous on $X \times \Lambda$, due to the well-known Analysis theorem L is uniformly continuous on $X \times \Lambda$: for every $\epsilon > 0$ there exists $\delta(\epsilon) > 0$ such that

$$|x - x'| + |\lambda - \lambda'| \leq \delta(\epsilon) \Rightarrow |L(x,\lambda) - L(x',\lambda')| \leq \epsilon \quad [4)] \tag{3.4.4}$$

[4)] for those not too familiar with Analysis, I wish to stress the difference between the usual continuity and the uniform continuity: continuity of L means that given $\epsilon > 0$ *and a point* (x,λ), it is possible to choose $\delta > 0$ such that (3.4.4) is valid; the corresponding δ may depend on (x,λ), not only on ϵ. Uniform continuity means that this positive δ may be chosen as a function of ϵ only. The fact that a continuous on a compact set function automatically is uniformly continuous on the set is one of the most useful features of compact sets

In particular,

$$|x - x'| \leq \delta(\epsilon) \Rightarrow |L(x, \lambda) - L(x'\lambda)| \leq \epsilon,$$

whence, of course, also

$$|x - x'| \leq \delta(\epsilon) \Rightarrow |\overline{L}(x) - \overline{L}(x')| \leq \epsilon,$$

so that the function $\overline{L}$ is continuous on X. Similarly, $\underline{L}$ is continuous on Λ. Taking in account that X and Λ are compact sets, we conclude that the problems (I) and (II) are solvable.

(ii) is the essence of the matter; here, of course, the entire construction heavily exploits convexity-concavity of L.

0^0. To prove (ii), we first establish the following statement, which is important by its own right:

Lemma 3.4.1 [Minmax Lemma] *Let X be a convex compact set and $f_0, ..., f_N$ be a collection of $N+1$ convex and continuous functions on X. Then the minmax*

$$\min_{x \in X} \max_{i=0,...,N} f_i(x) \tag{3.4.5}$$

of the collection is equal to the minimum in $x \in X$ of certain convex combination of the functions: there exist nonnegative μ_i, $i = 0, ..., N$, with unit sum such that

$$\min_{x \in X} \max_{i=0,...,N} f_i(x) = \min_{x \in X} \sum_{i=0}^{N} \mu_i f_i(x)$$

Remark 3.4.1 Minimum of *every* convex combination of a collection of *arbitrary* functions is $\leq$ the minmax of the collection; this evident fact can be also obtained from (3.4.2) as applied to the function

$$M(x, \mu) = \sum_{i=0}^{N} \mu_i f_i(x)$$

on the direct product of X and the standard simplex

$$\Delta = \{\mu \in \mathbf{R}^{N+1} \mid \mu \geq 0, \sum_i \mu_i = 1\}.$$

The Minmax Lemma says that if f_i are convex and continuous on a convex compact set X, then the indicated inequality is in fact equality; you can easily verify that this is nothing but the claim that the function M possesses a saddle point. Thus, the Minmax Lemma is in fact a particular case of the Sion-Kakutani Theorem; we are about to give a direct proof of this particular case of the Theorem and then to derive the general case from this particular one.

Proof of the Minmax Lemma. Consider the optimization program

$$(S) \quad \min_{t,x} \{t : f_0(x) - t \leq 0, \, f_1(x) - t \leq 0, ..., f_N(x) - t \leq 0, \, x \in X\}.$$

This clearly is a convex program with the optimal value

$$t^* = \min_{x \in X} \max_{i=0,...,N} f_i(x)$$

(note that (t, x) is feasible solution for (S) if and only if $x \in X$ and $t \geq \max_{i=0,...,N} f_i(x)$). The problem clearly satisfies the Slater condition and is solvable (since X is compact set and f_i, $i = 0, ..., N$, are continuous on X; therefore their maximum also is continuous on X and thus attains its minimum on the compact set X). Let (t^*, x^*) be an optimal solution to the

problem. According to Theorem 3.2.6, there exists $\lambda^* \geq 0$ such that $((t^*, x^*), \lambda^*)$ is a saddle point of the corresponding Lagrange function

$$L(t, x; \lambda) = t + \sum_{i=0}^{N} \lambda_i (f_i(x) - t) = t(1 - \sum_{i=0}^{N} \lambda_i) + \sum_{i=0}^{N} \lambda_i f_i(x),$$

and the value of this function at $((t^*, x^*), \lambda^*)$ is equal to the optimal value in (S), i.e., to t^*. Now, since $L(t, x; \lambda^*)$ attains its minimum in (t, x) over the set $\{t \in \mathbf{R}, x \in X\}$ at (t^*, x^*), we should have

$$\sum_{i=0}^{N} \lambda_i^* = 1$$

(otherwise the minimum of L in (t, x) would be $-\infty$). Thus,

$$[\min_{x \in X} \max_{i=0,\ldots,N} f_i(x) =] \quad t^* = \min_{t \in \mathbf{R}, x \in X} \left[t \times 0 + \sum_{i=0}^{N} \lambda_i^* f_i(x) \right],$$

so that

$$\min_{x \in X} \max_{i=0,\ldots,N} f_i(x) = \min_{x \in X} \sum_{i=0}^{N} \lambda_i^* f_i(x)$$

with some $\lambda_i^* \geq 0$, $\sum_{i=0}^{N} \lambda_i^* = 1$, as claimed. ∎

From the Minmax Lemma to the Sion-Kakutani Theorem. We should prove that the optimal values in (I) and (II) (which, by (i), are well defined reals) are equal to each other, i.e., that

$$\inf_{x \in X} \sup_{\lambda \in \Lambda} L(x, \lambda) = \sup_{\lambda \in \Lambda} \inf_{x \in X} L(x, \lambda).$$

We know from (3.4.4) that the first of these two quantities is greater than or equal to the second, so that all we need is to prove the inverse inequality. For me it is convenient to assume that the right quantity (the optimal value in (II)) is 0, which, of course, does not restrict generality; and all we need to prove is that the left quantity – the optimal value in (I) – cannot be positive.

1^0. What does it mean that the optimal value in (II) is zero? When it is zero, then the function $\underline{L}(\lambda)$ is nonpositive for every λ, or, which is the same, the convex continuous function of $x \in X$ – the function $L(x, \lambda)$ – has nonpositive minimal value over $x \in X$. Since X is compact, this minimal value is achieved, so that the set

$$X(\lambda) = \{x \in X \mid L(x, \lambda) \leq 0\}$$

is nonempty; and since X is convex and L is convex in $x \in X$, the set $X(\lambda)$ is convex (as a level set of a convex function, Proposition 2.1.4). Note also that the set is closed (since X is closed and $L(x, \lambda)$ is continuous in $x \in X$).

2^0. Thus, if the optimal value in (II) is zero, then the set $X(\lambda)$ is a nonempty convex compact set for every $\lambda \in \Lambda$. And what does it mean that the optimal value in (I) is nonpositive? It means exactly that there is a point $x \in X$ where the function $\overline{L}$ is nonpositive, i.e., the point $x \in X$ where $L(x, \lambda) \leq 0$ for all $\lambda \in \Lambda$. In other words, to prove that the optimal value in (I) is nonpositive is the same as to prove that *the sets $X(\lambda)$, $\lambda \in \Lambda$, have a point in common.*

3^0. With the above observations we see that the situation is as follows: we are given a family of closed nonempty convex subsets $X(\lambda)$, $\lambda \in \Lambda$, of a compact set X, and we should prove that these sets have a point in common. To this end, in turn, it suffices to prove that every *finite* number of sets from our family have a point in common (to justify this claim, I can refer to the Helley Theorem II, which gives us much stronger result: to prove that all $X(\lambda)$ have a point in common, it suffices to prove that every $(n + 1)$

sets of this family, n being the affine dimension of X, have a point in common). Let $X(\lambda_0), ..., X(\lambda_N)$ be $N+1$ sets from our family; we should prove that the sets have a point in common. In other words, let

$$f_i(x) = L(x, \lambda_i),\ i = 0, ..., N;$$

all we should prove is that there exists a point x where all our functions are nonpositive, or, which is the same, that the minmax of our collection of functions – the quantity

$$\alpha \equiv \min_{x \in X} \max_{i=1,...,N} f_i(x)$$

– is nonpositive.

The proof of the inequality $\alpha \leq 0$ is as follows. According to the Minmax Lemma (which can be applied in our situation – since L is convex and continuous in x, all f_i are convex and continuous, and X is compact), α is the minimum in $x \in X$ of certain convex combination $\phi(x) = \sum_{i=0}^{N} \nu_i f_i(x)$ of the functions $f_i(x)$. We have

$$\phi(x) = \sum_{i=0}^{N} \nu_i f_i(x) \equiv \sum_{i=0}^{N} \nu_i L(x, \lambda_i) \leq L(x, \sum_{i=0}^{N} \nu_i \lambda_i)$$

(the last inequality follows from concavity of L in λ; this is the only – and crucial – point where we use this assumption). We see that $\phi(\cdot)$ is majorated by $L(\cdot, \lambda)$ for a properly chosen λ; it follows that the minimum of ϕ in $x \in X$ – and we already know that this minimum is exactly α – is nonpositive (recall that the minimum of L in x is nonpositive for every λ). ∎

Semi-Bounded case. The next theorem lifts the assumption of boundedness of X and Λ in Theorem 3.4.2 – now only one of these sets should be bounded – at the price of some weakening of the conclusion.

Theorem 3.4.3 [Swapping min and max in convex-concave saddle point problem (Sion-Kakutani)] *Let X and Λ be convex sets in $\mathbf{R}^n$ and $\mathbf{R}^m$, respectively, with X being compact, and let*

$$L(x, \lambda) : X \times \Lambda \to \mathbf{R}$$

be a continuous function which is convex in $x \in X$ for every fixed $\lambda \in \Lambda$ and is concave in $\lambda \in \Lambda$ for every fixed $x \in X$. Then

$$\inf_{x \in X} \sup_{\lambda \in \Lambda} L(x, \lambda) = \sup_{\lambda \in \Lambda} \inf_{x \in X} L(x, \lambda). \tag{3.4.6}$$

Proof. By general theory, in (3.4.6) the left hand side is $\geq$ the right hand side, so that there is nothing to prove when the right had side is $+\infty$. Assume that this is not the case. Since X is compact and L is continuous in $x \in X$, $\inf_{x \in X} L(x, \lambda) > -\infty$ for every $\lambda \in \Lambda$, so that the left hand side in (3.4.6) cannot be $-\infty$; since it is not $+\infty$ as well, it is a real, and by shift, we can assume w.l.o.g. that this real is 0:

$$\sup_{\lambda \in \Lambda} \inf_{x \in X} L(x, \lambda) = 0.$$

All we need to prove now is that the left hand side in (3.4.6) is nonpositive. Assume, on the contrary, that it is positive and thus is $> c$ with some $c > 0$. Then for every $x \in X$ there exists $\lambda_x \in \Lambda$ such that $L(x, \lambda_x) > c$. By continuity, there exists a neighborhood V_x of x in X such that $L(x', \lambda_x) \geq c$ for all $x' \in V_x$. Since X is compact, we can find finitely many points $x_1, ..., x_n$ in X such that the union, over $1 \leq i \leq n$, of V_{x_i} is exactly X, implying that $\max_{1 \leq i \leq n} L(x, \lambda_{x_i}) \geq c$ for every $x \in X$. Now let $\bar{\Lambda}$ be the convex hull of $\{\lambda_{x_1}, ..., \lambda_{x_n}\}$, so that $\max_{\lambda \in \bar{\Lambda}} L(x, \lambda) \geq c$ for every $x \in X$. Applying to L and the convex *compact* sets $X, \bar{\Lambda}$ Theorem 3.4.2, we get the equality in the following chain:

$$c \leq \min_{x \in X} \max_{\lambda \in \bar{\Lambda}} L(x, \lambda) = \max_{\lambda \in \bar{\Lambda}} \min_{x \in X} L(x, \lambda) \leq \sup_{\lambda \in \Lambda} \min_{x \in X} L(x, \lambda) = 0,$$

which is a desired contradiction (recall that $c > 0$). ∎

Slightly strengthening the premise of Theorem 3.4.3, we can replace (3.4.6) with existence of a saddle point:

Theorem 3.4.4 [Existence of saddle point in convex-concave saddle point problem (Sion-Kakutani, Semi-Bounded case)] *Let X and Λ be closed convex sets in $\mathbf{R}^n$ and $\mathbf{R}^m$, respectively, with X being compact, and let*

$$L(x,\lambda) : X \times \Lambda \to \mathbf{R}$$

be a continuous function which is convex in $x \in X$ for every fixed $\lambda \in \Lambda$ and is concave in $\lambda \in \Lambda$ for every fixed $x \in X$. Assume that for every $a \in \mathbf{R}$, there exist a collection $x_1^a, ..., x_{n_a}^a \in X$ such that the set

$$\{\lambda \in \Lambda : L(x_i^a, \lambda) \geq a \, 1 \leq i \leq n_a\}$$

is bounded[5]*. Then L has saddle points on $X \times \Lambda$.*

Proof. Since X is compact and L is continuous, the function $\underline{L}(\lambda) = \min_{x \in X} L(x,\lambda)$ is real-valued and continuous on Λ. Further, for every $a \in \mathbf{R}$, the set $\{\lambda \in \Lambda : \underline{L}(\lambda) \geq a\}$ clearly is contained in the set $\{\lambda : L(x_i^a, \lambda) \geq a, \, 1 \leq i \leq n_a\}$ and thus is bounded. Thus, $\underline{L}(\lambda)$ is a continuous function on a closed set Λ, and the level sets $\{\lambda \in \Lambda : \underline{L}(\lambda) \geq a\}$ are bounded, implying that $\underline{L}$ attains its maximum on Λ. Invoking Theorem 3.4.3, it follows that $\inf_{x \in X}[\overline{L}(x) := \sup_{\lambda \in \Lambda} L(x,\lambda)]$ is finite, implying that the function $\overline{L}(\cdot)$ is not $+\infty$ identically in $x \in X$. Since L is continuous, $\overline{L}$ is lower semicontinuous. Thus, $\overline{L} : X \to \mathbf{R} \cup \{+\infty\}$ is a lower semicontinuous proper (i.e., not identically $+\infty$) function on X; since X is compact, $\overline{L}$ attains its minimum on X. Thus, both problems $\max_{\lambda \in \Lambda} \underline{L}(\lambda)$ and $\min_{x \in X} \overline{L}(x)$ are solvable, and the optimal values in the problem are equal by Theorem 3.4.3. Invoking Theorem 3.4.1, L has a saddle point. ∎

[5] this definitely is the case true when $L(\bar{x}, \lambda)$ is coercive in λ for some $\bar{x} \in X$, meaning that the sets $\{\lambda \in \Lambda : L(\bar{x}, \lambda) \geq a\}$ are bounded for every $a \in \mathbf{R}$, or, equivalently, whenever $\lambda_i \in \Lambda$ and $\| \lambda_i \|_2 \to \infty$ as $i \to \infty$, we have $L(\bar{x}, \lambda_i) \to -\infty$ as $i \to \infty$.

Lecture 4

Optimality Conditions

This Lecture, the last in the theoretical part of the course, is devoted to optimality conditions in general-type Mathematical Programming programs

$$(P) \quad \min_x \{f(x) : g(x) \equiv (g_1(x), g_2(x), ..., g_m(x)) \leq 0,\ h(x) = (h_1(x), ..., h_k(x)) = 0,\ x \in X\}.$$

The question we are interested in is as follows:

- Assume that we are given a feasible solution x^* to (P). What are the conditions (necessary, sufficient, necessary and sufficient) for x^* to be optimal?

We, as it is normally the case in Mathematical Programming, intend to answer this question under the following assumptions on the data:

- A. x^* is an interior point of the domain of the problem X;
- B. The functions $f, g_1, ..., g_m, h_1, ..., h_k$ are smooth at x^* (at least once continuously differentiable in a neighbourhood of the point; when necessary, we shall require more smoothness)

Let me stress that we are <u>not</u> going to impose on the problem structural assumptions like convexity, in contrast to what was done in the previous Lecture.

Before coming to "technical" considerations, let us discuss the following "philosophical" questions:

- Conditions of what kind are we interested in?
- Why are we interested in these conditions?

The answer to the first question is as follows: we are interested in <u>local</u> and <u>verifiable</u> optimality conditions. Locality means that the conditions should be expressed in terms of local properties of the data – in terms of the values and derivatives (of first, second,... order) of the functions $f, g_1, ..., g_m, h_1, ..., h_k$ at x^*. Verifiability means that given the values and the derivatives of the indicated functions at x^*, we should be able to verify efficiently whether the condition is or is not satisfied.

The outlined – quite reasonable – requirements to the conditions to be derived lead to rather unpleasant consequences:

> *What we may hope for are <u>necessary</u> conditions for optimality of x^* and <u>sufficient</u> conditions for <u>local</u> optimality of x^*, and not <u>sufficient</u> conditions for <u>global</u> optimality of x^*.*

Let me explain, first, what is meant in the above claim – what does it mean "local" and "global" optimality, and, second, why the claim is true.

<u>Global</u> optimality of x^* is nothing but the actual optimality: x^* is a feasible solution to (P) with the best possible value of the objective – the one which is $\leq$ the value of the objective at any other feasible solution to the problem. In contrast to this, <u>local</u> optimality of x^* means that x^* is feasible solution which is not worse, from the viewpoint of the values of the objective, than other feasible solutions <u>close</u> enough to x^*. The formal definition is as follows:

A feasible solution x^* to (P) is called locally optimal, if there exists a neighbourhood U of x^* such that x^* is optimal solution of the "restricted to U" version of (P), i.e., if

$$x \in U, g(x) \leq 0, h(x) = 0 \Rightarrow f(x) \geq f(x^*).$$

(note that in the last relation we skip the inclusion $x \in X$ in the premise of the implication; this is because we have assumed that x^* is an interior point of X, so that shrinking, if necessary, U, we always can make it part of X and thus make the inclusion $x \in X$ a consequence of the inclusion $x \in U$).

In the convex case local optimality is the same as the global one (this follows from Theorem 2.5.1 combined with the fact that the feasible set of a convex program is convex). In the general case these two notions are different – a globally optimal solution is, of course, a locally optimal one, but not vice versa: look at something like the problem

$$\min_x \left\{ f(x) = 0.1x^2 + \sin^2 x \right\};$$

here all points $x_k = \pi k$ are local minimizers of the objective, but only one of them – 0 – is its global minimizer.

Note that since a globally optimal solution for sure is a locally optimal one, the necessary condition for local optimality (and these are all necessary optimality conditions to be discussed) are necessary for global optimality as well.

Now, why it is true that in the general case it is impossible to point out local sufficient condition for global optimality, it also is clear: because local information on a function f at a local minimizer x^* of the function does not allow to understand that the minimizer is only local and not a global one. Indeed, let us take the above f and $x^* = \pi$; this is only local, not global, minimizer of f. At the same time we can easily change f outside a neighbourhood of x^* and make x^* the global minimizer of the updated function (draw the graph of f to see it). Note that we can easily make the updated function $\bar{f}$ as smooth as we wish. Now, local information – value and derivatives at x^* – on f and on the updated function $\bar{f}$ are the same, since the functions coincide with each other in a neighbourhood of x^*. It follows that there is no test which takes on input local information on the problem at x^* and correctly reports on output whether x^* is or is not a global minimizer of the objective, even if we assume the objective to be very smooth. Indeed, such a test is unable to distinguish the above f and $\bar{f}$ and is therefore enforced, being asked once about f and once about $\bar{f}$, report both times the same answer; whatever is the answer, in one of these two cases it is false!

The difficulty we have outlined is intrinsic for nonconvex optimization: not only there does not exist "efficient local test" for global optimality; there also does not exist, as we shall see in our forthcoming lectures, an efficient algorithm capable to approximate global minimizer of a general-type Mathematical Programming problem, even one with very smooth data.

In view of this unpleasant and unavoidable feature of general-type Mathematical Programming problems, the answer to the second of the outlined questions – what for we use Optimality conditions in Mathematical Programming – is not as optimistic as we would wish it to be. As far as conditions for global optimality are concerned, we may hope for necessary optimality conditions only; in other words, we may hope for a test which is capable to say that what we have is not a globally optimal solution. Since there is no (local) *sufficient* condition for global optimality, we have no hope to design a local test capable to say that what we have is the "actual" – global – solution to the problem. The best we may hope for in this direction is a sufficient condition for local optimality, i.e., a local test capable to say that what we have cannot be improved by small modifications.

The pessimism caused by the above remarks has, however, its bounds. A necessary optimality condition is certain relation which must be satisfied at optimal solution. If we are clever enough to generate – on paper or algorithmically – all candidates x^* which satisfy this relation, and if the list of these candidates turns out to be finite, then we may hope that looking through the list and choosing in it the best, from the viewpoint of the objective, feasible solution, we eventually shall find the globally optimal solution (given that it exists). Needless to say, the outlined possibility is met only in "simple particular cases", but already these cases sometimes are extremely important (we shall discuss an example of this type at the end of this Lecture). Another way to utilize necessary and/or sufficient conditions for local

optimality is to use them as "driving force" in optimization algorithms. Here we generate a sequence of approximate solutions and subject them to the test for local optimality given by our optimality condition. If the current iterate passes the test, we terminate with a locally optimal solution to the problem; if it is not the case, then the optimality condition (which is violated at the current iterate) normally says to us how to update the iterate in order to reduce "violation" of the condition. As a result of these sequential updatings, we get a sequence of iterates which, under reasonable assumptions, can be proved to converge to a locally optimal solution to the problem. As we shall see in the forthcoming lectures, this idea underlies all traditional computational methods of Mathematical Programming. Of course, with this scheme it in principle is impossible to guarantee convergence to globally optimal solution (imagine that we start at a locally optimal solution which is not a globally optimal one; with the outlined scheme, we shall terminate immediately!) Although this is a severe drawback of the scheme, it does not kill the traditional "optimality-conditions-based" methods. First, it may happen that we are lucky and there are no "false" local solutions – the only local solution is the global one; then the above scheme will approximate the actual solution (although we never will know that it is the case...) Second, in many practical situations we are interested in improving a given initial solution to the problem rather than in finding the "best possible" solution, and the traditional methods normally allow to achieve this restricted goal.

Now let us pass from the "motivation preamble" to the mathematics of optimality conditions. There are two kinds of them – conditions utilizing the first-order information of the objective and the constraints at x^* only (the values and the gradients of these functions), and second-order conditions using the second order derivatives as well. As we shall see, all first-order optimality conditions are only *necessary* for optimality. Among the second-order optimality conditions, there are both necessary and sufficient for local optimality.

4.1 First Order Optimality Conditions

The idea of the first order optimality conditions is extremely simple. Consider an optimization problem (P), and let x^* be a feasible solution to the problem. To derive a necessary condition for local optimality of x^* is the same as to find a consequence of the fact that x^* is locally optimal; such a consequence is, of course, a necessary optimality condition. Thus, assume that x^* is locally optimal for (P), and let us try to guess what can be derived from this fact. The most straightforward idea is as follows: let us approximate the objective and the constraints of the actual problem (P) in a neighbourhood of x^* by "simple" functions, thus coming to "approximating problem" $(\widehat{P})$. We may hope that if the approximation is good enough, that the local property of (P) we are interested in – the property that x^* is a locally optimal solution to (P) – will be inherited by $(\widehat{P})$. If

- (A) $(\widehat{P})$ is that simple that we can say "constructively" what does it mean that x^* is locally optimal for $(\widehat{P})$,

  and

- (B) we can prove that our hypothesis

  > *"if x^* is locally optimal for (P), it is locally optimal for $(\widehat{P})$ as well"*

  is true,

then the condition given by (A) will be necessary for local optimality of x^* in (P).

There is, basically, only one natural way to implement this idea, given that we are interested in the first order optimality conditions and, consequently, that $\widehat{P})$ should be posed in terms of the values and the gradients of the original objective and constraints at x^* only. This way is to linearize the original objective and constraints at x^* and to make the resulting affine functions the objective and the constraints of $(\widehat{P})$. The linearizations in question are

$$\begin{array}{rcl} \bar{f}(x) & = & f(x^*) + (x - x^*)^T \nabla f(x^*), \\ \bar{g}_i(x) & = & g_i(x^*) + (x - x^*)^T \nabla g_i(x^*),\ i = 1, ..., m, \\ \bar{h}_i(x) & = & h_i(x^*) + (x - x^*)^T \nabla h_j(x^*),\ j = 1, ..., k, \end{array}$$

which results in the *Linear Programming* problem $(\widehat{P})$:

$$\begin{array}{ll}(\widehat{P}):&\\ \min & f(x^*)+(x-x^*)^T\nabla f(x^*)\\ \text{s.t.}&\\ & g_i(x^*)+(x-x^*)^T\nabla g_i(x^*) \leq 0,\ i=1,...,m\\ & (x-x^*)^T\nabla h_j(x^*) = 0,\ j=1,...,k\end{array}$$

(we omit $h_j(x^*)$ – they are zeros, since x^* is feasible).

Now, when x^* is an optimal solution to the LP program $(\widehat{P})$, it is said by the Linear Programming Duality Theorem. Since we have not established the theorem for the particular form of the LP program we are interested in (the one with equality constraints and not inequality constraints only), it makes sense to derive the required optimality condition explicitly from the source of the LP Duality Theorem – from the Homogeneous Farkas Lemma.

Assume that x^* (which is clearly feasible for $(\widehat{P})$ – recall that x^* is feasible for (P)) is optimal for $(\widehat{P})$. Let $I(x^*)$ be the set of indices of all inequality constraints of (P) which are active (satisfied as equalities) at x^*, and consider the set

$$K=\{d \mid d^T\nabla g_i(x^*)\leq 0,\ i\in I(x^*), d^T\nabla h_j(x^*)=0,\ j=1,...,k\}.$$

It is absolutely clear that if $d\in K$, then all vectors $x_t=x^*+td$ corresponding to small enough positive t are feasible for $(\widehat{P})$. Since x^* is optimal for the latter problem, we should have

$$f(x^*)+(x_t-x^*)^T\nabla f(x^*)\geq f(x^*)$$

for the indicated t, whence $d^T\nabla f(x^*)\geq 0$. Thus,

(*) *if x^* is optimal for $(\widehat{P})$, then $d^T\nabla f(x^*)\geq 0$ for all $d\in K$;*

in fact "if ... then ..." here can be replaced with "if and only if" (why?).

Now, by the Homogeneous Farkas Lemma (Lecture 3) statement (*), in turn, is equivalent to the possibility of representation

$$\nabla f(x^*) = -\sum_{i\in I(x^*)}\lambda_i^*\nabla g_j(x^*) - \sum_{j=1}^k \mu_j\nabla h_j(x^*) \tag{4.1.1}$$

with some nonnegative λ_i^* and some real μ_j^*. To see it, note that K is exactly the polyhedral cone

$$\{d \mid d^T\nabla g_i(x^*)\leq 0,\ i\in I(x^*),\ d^T\nabla h_j(x^*)\leq 0, d^T(-\nabla h_j(x^*))\leq 0,\ j=1,...,k\},$$

and (*) says that the vector $\nabla f(x^*)$ has nonnegative inner products with all vectors from K, i.e., with all vectors which have nonnegative inner products with the vectors from the finite set

$$A=\{-\nabla g_i(x^*),\ i\in I(x^*), \pm\nabla h_j(x^*),\ j=1,...,k\}.$$

By the Homogeneous Farkas Lemma this is the case if and only if $\nabla f(x^*)$ is combination with nonnegative coefficients of the vectors from A:

$$\nabla f(x^*) = -\sum_{i\in I(x^*)}\lambda_i^*\nabla g_i(x^*) + \sum_{j=1}^k[\mu_{j,+}^*-\mu_{j,j}^*]\nabla h_j(x^*)$$

with nonnegative $\lambda_j^*, \mu_{j,+}^*, \mu_{j,-}^*$. And to say that $\nabla f(x^*)$ is representable in the latter form is the same as to say that it is representable as required in (4.1.1).

Now, λ_i^* to the moment are defined for $i\in I(x^*)$ only. We loose nothing when defining $\lambda_i^*=0$ for $i\notin I(x^*)$ and treating the right hand side sum in (4.1.1) as the sum over all $i=1,...,m$. Note also that now we have the complementary slackness relations $\lambda_i^* g_i(x^*)=0$, $i=1,...,m$.

We have established the following conditional statement:

Proposition 4.1.1 *Let x^* be locally optimal for (P) and such that the hypothesis* (B) *takes place: x^* remains optimal also for the linearized LP program $(\widehat{P})$. Then there exist nonnegative λ_i^* and real μ_j^* such that*

$$\begin{array}{rcll} \lambda_i^* g_i(x^*) & = & 0,\ i=1,...,m & \textit{[complementary slackness]} \\ \nabla f(x^*) + \sum\limits_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum\limits_{j=1}^{k} \mu_j^* \nabla h_j(x^*) & = & 0 & \textit{[KKT Equation]} \end{array} \tag{4.1.2}$$

The property of x^* to be feasible for (P) and to satisfy the condition "there exist nonnegative λ_i^* and real μ_j^* such that ..." from the above Proposition is called the Karush-Kuhn-Tucker Optimality Condition; we already know the version of this condition related to the case of inequality constrained problems. The point x^* which satisfies the KKT Optimality Condition is called a KKT point of (P) (sometimes this name is used for the pair $(x^*; \lambda^*, \mu^*)$, i.e., for the point x^* along with the certificate that it satisfies the KKT condition).

From the above discussion it follows that all we may hope for is that the KKT condition is necessary for local optimality of x^*; Proposition 4.1.2 says that this indeed is the case, but under implicit additional assumption "x^* remains...". The problem, consequently, is to convert this implicit assumption into something verifiable or to eliminate the assumption at all. The latter, unfortunately, is impossible, as it is seen from the following elementary example (where the problem is even convex):

$$\min_x \left\{ f(x) \equiv x : g_1(x) \equiv x^2 \le 0 \right\}.$$

The optimal solution to the problem (in fact – the only feasible solution to it) is $x^* = 0$. However, $x^* = 0$ is not a KKT point of the problem – it is impossible to find nonnegative λ_1^* such that

$$\nabla f(0) + \lambda_1^* \nabla g_1(0) \equiv 1 + \lambda_1^* \times 0 = 0.$$

Thus, we indeed need some "regularity assumption" to make the KKT Optimality Condition necessary for local optimality. The most general regularity assumption of this type is called "Qualification of Constraints". Let us look what does it mean.

Qualification of constraints actually says that the *feasible set* of the actual problem (P) should approximate the feasible set of the linearized problem $(\widehat{P})$ in a neighbourhood of x^* "up to the highest-order terms in $|x - x^*|$", similarly to what is the case with the data of the problems. To give the precise definition, let us agree to write

$$\theta(t) = o(t^s)$$

(θ is a function on the nonnegative ray, $s > 0$), if $\theta(t)t^{-s} \to 0$ as $t \to +0$ and $\theta(0) = 0$; this is one of standard Calculus conventions. And we say that *problem (P) satisfies the Qualification of Constraints property at feasible solution x^**, if there exists function $\theta(t) = o(t)$ such that

> *for every feasible solution x to the linearized problem $(\widehat{P})$ there exists a feasible solution x' to the actual problem (P) such that*
>
> $$|x - x'| \le \theta(|x - x^*|)$$
>
> *– the distance between x and x' goes to zero faster than the distance from x to x^* as $x \to x^*$.*

The Qualification of Constraints condition roughly speaking says that the feasible set of the linearized problem $(\widehat{P})$ cannot be (locally, of course) "much wider" than the feasible set of (P): for every x close to x^* and feasible for $(\widehat{P})$ there exists x' "very close" to x and feasible for (P). Note that in the above "bad example" the situation is opposite: the feasible set of $(\widehat{P})$ is the entire axis (since the constraint in the linearized problem is $0 \times x \le 0$), which is a "much wider" set, even locally, than the feasible set $\{0\}$ of (P).

It is easily seen that under the Qualification of Constraints assumption local optimality of x^* for (P) implies global optimality of x^* for $(\widehat{P})$, so that this assumption makes the KKT Optimality condition necessary for optimality:

Proposition 4.1.2 *Let x^* be locally optimal for (P) and let (P) satisfy the Qualification of Constraint assumption at x^*. Then x^* is optimal for $(\widehat{P})$ and, consequently, is a KKT point of (P).*

Proof. Let x^* be locally optimal for (P) and let the Qualification of Constraints take place; we should prove that then x^* is optimal for $(\widehat{P})$. Assume, on contrary, that x^* is not optimal for $(\widehat{P})$. Since x^* clearly is feasible for $(\widehat{P})$, non-optimality of x^* for the latter problem means that there exists a feasible solution $\bar{x}$ to $(\widehat{P})$ with less value of the linearized objective $f(x^*) + (x - x^*)^T \nabla f(x^*)$ than the value of this objective at x^*. Setting $d = \bar{x} - x^*$, we therefore obtain

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

Now let

$$x_t = x^* + t(\bar{x} - x^*),\ 0 \le t \le 1.$$

The points x_t are convex combinations of two feasible solutions to $(\widehat{P})$ and therefore also are feasible solutions to the latter problem (it is an LP program!) By Qualification of Constraints, there exist feasible solutions x'_t to the actual problem (P) such that

$$|x_t - x'_t| \le \theta(|x_t - x^*|) = \theta(t|\bar{x} - x^*|) \equiv \theta(tq),\ q = |\bar{x} - x^*|, \tag{4.1.3}$$

with $\theta(t) = o(t)$. Now, f is continuously differentiable in a neighbourhood of x^* (this is our once for ever assumption made in the beginning of the Lecture). It follows – this is a well-known fact from Analysis (an immediate consequence of the Lagrange Mean Value Theorem) – that f is locally Lipschitz continuous at x^*: there exists a neighbourhood U of x^* and a constant $C < \infty$ such that

$$|f(x) - f(y)| \le C|x - y|,\ x, y \in U. \tag{4.1.4}$$

As $t \to +0$, we have $x_t \to x^*$, and since $|x'_t - x_t| \le \theta(tq) \to 0$, $t \to 0$, x'_t also converges to x^* as $t \to 0$; in particular, both x_t and x'_t belong to U for all small enough positive t. Besides this, from local optimality of x^* and the fact that x'_t converges to x^* as $t \to +0$ and is feasible for (P) for all t we conclude that

$$f(x'_t) \ge f(x^*)$$

for all small enough positive t. It follows that for small positive t we have

$$\begin{array}{rcll} 0 & \le & t^{-1}[f(x'_t) - f(x^*)] & \\ & \le & t^{-1}[f(x_t) - f(x^*)] + t^{-1}[f(x'_t) - f(x_t)] & \\ & \le & t^{-1}[f(x_t) - f(x^*)] + t^{-1}C|x'_t - x_t| & [\text{see (4.1.4)}] \\ & \le & t^{-1}[f(x_t) - f(x^*)] + t^{-1}C\theta(tq) & [\text{see (4.1.3)}] \\ & = & \frac{f(x^* + td) - f(x^*)}{t} + t^{-1}C\theta(tq). & \end{array}$$

As $t \to 0$, the last expression in the chain tends to $d^T \nabla f(x^*) < 0$ (since $\theta(tq) = o(t)$), while the chain itself says that the expression is nonnegative. This is the desired contradiction. ∎

Proposition 4.1.2 is very close to tautology: the question was when the KKT condition is necessary for local optimality, and the answer we have now – that it for sure is the case when (P) satisfies the Qualification of Constraints assumption at x^*. If we can gain something from this answer, this something is indeed very small – we do not know how to certify that the Qualification of Constraints takes place. There is one trivial case – the one when the constraints of (P) are linear; in this case the feasible set of the linearized problem is even not close, but simply coincides with the feasible set of the actual problem (in fact it suffices to assume linearity of the active at x^* constraints only; then the feasible sets of (P) and $(\widehat{P})$ coincide with each other in a neighbourhood of x^*, which is quite sufficient for Constraint Qualification).

Among the more general certificates – sufficient conditions – for the Qualification of Constraints [1)] the most frequently used is the assumption of regularity of x^* for (P), which is the property as follows:

[1)] look how strange is what we are doing – we are discussing *sufficient condition* for something – namely, the Qualification of Constraints – which in turn is nothing but a *sufficient condition* to make something third – the KKT – *necessary condition* for local optimality. There indeed is something in human being, if he/she is capable to understand these "conditions for conditions" and to operate them!

Definition 4.1.1 *A feasible solution x of problem (P) is called regular, if the gradients of all active at x constraints of the problem are linearly independent.*

Why regularity of x^* implies Qualification of Constraints, it becomes clear from the following fundamental theorem of Analysis (this is one of the forms of the Implicit Function Theorem):

Theorem 4.1.1 *Let x^* be a point from $\mathbf{R}^n$ and $\phi_1, ..., \phi_l$ be $k \geq 1$ times continuously differentiable in a neighbourhood of x^* functions which are equal to 0 at x^* and are such that their gradients $\nabla\phi_i(x^*)$ at x^*, $i = 1, ..., l$, form a linearly independent set.*

Then there exists k times continuously differentiable along with its inverse substitution of argument

$$x = S(y)$$

which makes all the functions the coordinate ones, i.e. there exists

- *a neighbourhood X of the point x^* in $\mathbf{R}^n$*
- *a neighbourhood Y of the origin in $\mathbf{R}^n$*
- *a one-to-one mapping $y \mapsto S(y)$ of Y onto X which maps $y = 0$ to x^*: $S(0) = x^*$*

– such that

- (I) *S is k times continuously differentiable in Y, and its inverse $S^{-1}(x)$ is k times continuously differentiable in X;*
- (ii) *the functions*

  $$\psi_i(y) \equiv \phi_i(S(y))$$

  in Y are just the coordinate functions y_i, $i = 1, ..., l$.

Corollary 4.1.1 *Let $x^*, \phi_1, ..., \phi_l$ satisfy the premise of Theorem 4.1.1, let $q \leq l$, let X be the neighbourhood of x^* given by the Theorem, and let Φ be the solution set of the system*

$$\phi_i(x) \leq 0,\ i = 1, ..., q;\ \phi_i(x) = 0,\ i = q+1, ..., l.$$

Then there exists a neighbourhood $U \subset X$ of x^* such that the distance from a point $x \in U$ to Φ is bounded from above by constant times the norm of the "violation vector"

$$\delta(x) = \begin{pmatrix} \max\{\phi(x), 0\} \\ ... \\ \max\{\phi_q(x), 0\} \\ |\phi_{q+1}(x)| \\ ... \\ |\phi_l(x)| \end{pmatrix},$$

i.e., there exists a constant $D < \infty$ such that for every $x \in U$ there exists $x' \in \Phi$ with

$$|x - x'| \leq D|\delta(x)|. \tag{4.1.5}$$

Proof. Let V be a closed ball of positive radius r which is centered at the origin and is contained in Y. Since S is at least once continuously differentiable in a neighbourhood of the compact set V, its first order derivatives are bounded in V and therefore S is Lipschitz continuous in V with certain constant $D > 0$:

$$|S(y') - S(y'')| \leq D|y' - y''| \quad \forall y', y'' \in V.$$

Since S^{-1} is continuous and $S^{-1}(x^*) = 0$, there exists a neighbourhood $U \subset X$ of x^* such that S^{-1} maps this neighbourhood into V.

Now let $x \in U$, and consider the vector $y = S^{-1}(x)$. Due to the origin of U, this vector belongs to V, and due to the origin of S, the first l coordinates of the vector are exactly $\phi_i(x)$, $i = 1, ..., l$ (since $x = S(y)$, and we know that $\phi_i(S(y)) = y_i$, $i = 1, ..., l$). Now consider the vector y' with the coordinates

$$y'_i = \begin{cases} \min\{y_i, 0\}, & i = 1, ..., q \\ 0, & i = q+1, ..., l \\ y_i, & i = l+1, ..., n \end{cases}.$$

It is absolutely clear that

- (a) $|y'| \leq |y|$, so that $y' \in V$ along with y;
- (b) the first l entries in the vector $y' - y$ form the violation vector $\delta(x)$, and the remaining entries in $y' - y$ are zero, so that $|y' - y| = |\delta(x)|$.

Now let us set $x' = S(y')$. Since the first l coordinates of $y' = S^{-1}(x')$ are exactly $\phi_i(x')$, $i = 1, ..., l$, we see that the values of $\phi_1, ..., \phi_q$ at x' are nonpositive, and the values of the remaining ϕ's are zero, so that $x' \in \Phi$. On the other hand,

$$|x - x'| \equiv |S(y) - S(y')| \leq D|y - y'| = D|\delta(x)|$$

(we have used the Lipschitz continuity of S in V), as required. ∎

First Order Optimality Conditions. Now we are able to reach the first of our two targets – to get the First Order Optimality Conditions.

Theorem 4.1.2 [First Order Necessary Optimality Conditions in Mathematical Programming]
Consider optimization program (P), and let x^* be feasible solution to the problem. Assume that $f, g_1, ..., g_m, h_1, ..., h_k$ are continuously differentiable in a neighbourhood of x^* and that
– either all the constraints of (P) which are active at x^* are linear,
– or x^* is regular, i.e., the taken at x^* gradients of the constraints active at x^* form a linearly independent set.
Then the KKT Optimality Condition is necessary for x^* to be locally optimal solution to (P). Besides this, if x^* is regular and locally optimal solution to (P), then the Lagrange multipliers λ_i^*, μ_j^* certifying this latter fact are uniquely defined.

Proof. In view of Proposition 4.1.2, all we need is to verify that
(i) (P) satisfies the Qualification of Constraints assumption at x^* (this will imply that if x^* is locally optimal for (P), then it is a KKT point of the problem)
and
(ii) if x^* is regular and locally optimal for (P) (so that, by (i), it is a KKT point of the problem), then the corresponding Lagrange multipliers are uniquely defined.
(ii) is immediate: the Lagrange multipliers corresponding to the non-active at x^* inequality constraints must be 0 by complementary slackness, and the remaining Lagrange multipliers, from the KKT Equation, are the coefficients in the representation of $-\nabla f(x^*)$ as a linear combination of the taken at x^* gradients of the active constraints. When x^* is regular, the latter gradients are linearly independent, so that the coefficients in the above combination are uniquely defined.
Now let us verify (i). There is no problem to establish (i) for the case when all the constraints of (P) active at x^* are linear – here the Qualification of Constraints is evident. Thus, all we need is to assume that x^* is regular and to derive from this assumption the Qualification of Constraints property. To this end let $\{\phi_1, ..., \phi_l\}$ be the group comprised of the inequality constraints active at x (the first q functions of the group) and all the equality constraints (the remaining $l - q$ functions). This group, along with x^*, satisfies the premise in Corollary 4.1.1; according to the Corollary, there exists a neighbourhood U of x^* and a constant D such that

$$\forall x \in U \quad \exists x' : \quad |x - x'| \leq D|\delta(x)|,\ \phi_i(x') \leq 0,\ i = 1, ..., q;\ \phi_i(x') = 0,\ i = q+1, ..., l. \tag{4.1.6}$$

Besides this, there exists a neighbourhood W of x^* such that all the inequality constraints which are not active at x^* are satisfied in the entire W (indeed, all the constraint functions are continuous at x^*, so that the constraints nonactive at x^*, being strict inequalities at the point, indeed are satisfied in a neighbourhood of x^*). Now consider the mapping

$$x \mapsto x'(x)$$

defined as follows: for $x \in U$, $x'(x)$ is the vector x' given by (4.1.6), if the latter vector belongs to W; otherwise, same as in the case of $x \notin U$, let $x'(x) = x^*$. Note that with this definition of $x'(x)$, this latter vector always is a feasible solution to (P) (why?) Besides this, as $x \to x^*$, then the violation vector $\delta(x)$ clearly goes to 0, so that x' given by (4.1.6) also goes to x^* and therefore eventually becomes a vector from W; it follows that for all x close enough to x^* the vector $x'(x)$ is exactly the vector given by (4.1.6). Summarizing our observations, we come to the following conclusions:

> We have defined a mapping which puts into correspondence to an arbitrary $x \in \mathbf{R}^n$ a feasible solution $x'(x)$ to (P). This mapping is bounded, and in certain neighbourhood Q of x^* is such that
>
> $$|x'(x) - x| \le D|\delta(x)|. \tag{4.1.7}$$

Now assume that x is a feasible solution to the linearized problem $(\widehat{P})$. Note that the vector $\phi(x) = (\phi_1(x), ..., \phi_l(x))$ admits the representation

$$\phi(x) = \phi^{\rm lin}(x) + \phi^{\rm rem}(x),$$

where $\phi^{\rm lin}$ comes from the linearizations of the functions ϕ_i at x^* – i.e., from the constraint functions of $(\widehat{P})$, and $\phi^{\rm rem}$ comes from the remainders in the first order Taylor expansions of ϕ_i at x^*. *Since x is feasible for $(\widehat{P})$, the first q entries of $\phi^{\rm lin}(x)$ clearly are nonpositive, and remaining entries are equal to 0.* It follows that *if x is feasible for $(\widehat{P})$, then the norm of the violation vector $\delta(x)$ does not exceed the norm of the vector $\phi^{\rm rem}(x)$* (look at the definition of the violation vector), and the latter norm is $\le \theta(|x - x^*|)$ with certain $\theta(t) = o(t)$; indeed, the remainders in the first order Taylor expansions of continuously differentiable in a neighbourhood of x^* constraint functions are $o(|x - x^*|)$, x being the point where the remainders are evaluated. Combining this observation with (4.1.7), we conclude that there is a neighbourhood Z of x^* such that if $x \in Z$ is feasible for $(\widehat{P})$, then

$$|x'(x) - x| \le D|\delta(x)| \le D|\phi^{\rm rem}(x)| \le D\theta(|x - x^*|) \tag{4.1.8}$$

with some $\theta(t) = o(t)$. Outside Z the left hand side in the latter equation clearly is bounded from above by $D'|x - x^*|$ for some D' (recall that $x'(x)$ is bounded), so that, redefining $\theta(t)$ in an appropriate manner outside a neighbourhood of $t = 0$, we can ensure that (4.1.8) is valid for all x feasible for $(\widehat{P})$. Since $x'(x)$, by construction, is feasible for (P), (4.1.8) demonstrates that the Qualification of Constraints does hold. ∎

4.2 Second Order Optimality Conditions

A weak point of the first order optimality conditions is that they at best are only necessary, but not sufficient, for local optimality. Indeed, look at the simplest of these conditions, the one for smooth unconstrained minimization: the Fermat rule $\nabla f(x^*) = 0$. This condition does not distinguish between local minima, local maxima and "saddle points" (local minimum in some directions and local maximum in other directions) and therefore definitely is not sufficient for local optimality (excluding the case when we assume convexity of f). It follows that if we are interested in sufficient conditions for optimality, the conditions should exploit the second order information as well. Let us look what happens in the simplest case when the problem is unconstrained – it is

$$\min_x \{\phi(x) : x \in \mathbf{R}^n\},$$

and ϕ is twice continuously differentiable in a neighbourhood of a point x^* which we are interested to test for local optimality. When x^* is a local minimum of ϕ?

There are two related answers. The first is given by the necessary optimality condition "if x^* is local minimizer of ϕ, then the gradient $\nabla\phi(x^*)$ should vanish, and the Hessian $\nabla^2 f(x^*)$ should be positive semidefinite":

$$\begin{array}{rcll} x^* \text{ is local minimizer} & \Rightarrow & & \\ \nabla\phi(x^*) & = & 0 & \text{[Fermat rule – the first-order part]} \\ \nabla^2\phi(x^*) & \geq & 0 & \text{[i.e., } d^T\nabla^2\phi(x^*)d \geq 0 \quad \forall d - \\ & & & \text{the second-order part of the condition]} \end{array}$$

The second is the sufficient condition "if the gradient $\nabla\phi(x^*)$ vanishes and the Hessian $\nabla^2\phi(x^*)$ is positive definite, then x^* is local minimizer of f":

$$\nabla\phi(x^*) = 0 \quad \& \quad \nabla^2\phi(x^*) > 0 \text{ [i.e., } d^T\nabla^2\phi(x^*)d > 0 \quad \forall d \neq 0] \\ \Downarrow \\ x^* \text{ is local minimizer}$$

The proof of this well-known fact of Calculus is very easy: let us write down the second-order Taylor expansion of ϕ at x^* ($t > 0$, d is a unit vector: $|d| = 1$):

$$(*) \quad \phi(x^* + td) - \phi(x^*) = td^T\nabla\phi(x^*) + \frac{t^2}{2}d^T\nabla^2\phi(x^*)d + \theta(t) \quad [\theta(t) = o(t^2)].$$

If x^* is a local minimizer of ϕ, then the left hand side in this relation is nonnegative whenever $|d| = 1$ and t is small enough positive real; dividing both sides of the relation by t and passing to limit as $t \to +0$, we get $d^T\nabla\phi(x^*) \geq 0$ for all unit vectors d, which is possible if and only if $\nabla\phi(x^*) = 0$. Given this fact, dividing both sides of the relation by t^2 and passing to limit as $t \to +0$, we get $d^T\nabla^2\phi(x^*)d \geq 0$ for all unit (and then – for all) vectors d. Thus, we come to the necessary second-order optimality condition. To prove sufficiency of the sufficient second-order optimality condition, note that under this condition the linear in t term in the right hand side of (*) vanishes, and the right hand side itself can be rewritten as

$$(**) \quad \frac{t^2}{2}[d^T\nabla^2\phi(x^*)d + t^{-2}\theta(t)]$$

Due to the positive definiteness of $\nabla^2\phi(x^*)$, the first term in the parentheses is positive (and, of course, continuous) function on the unit sphere $\{d \mid |d| = 1\}$. Since the sphere is compact, this function attains its minimum on the sphere, so that its minimum is positive. It follows that the function $d^T\nabla^2\phi(x^*)d$ is bounded away from 0 on the unit sphere:

$$d^T\nabla^2\phi(x^*)d \geq \alpha > 0 \quad \forall d, |d| = 1.$$

Thus, the first term in the parentheses in (**) is $\geq \alpha > 0$. The second term tends to 0 as $t \to +0$, since $\theta(t) = 0(t^2)$. Thus, there exists $\delta > 0$ such that quantity in parentheses is strictly positive whenever $0 < t \leq \delta$. Now we see that the right hand side in (*) is positive whenever $|d| = 1$ and $0 < t \leq \delta$, and (*) says that x^* indeed is a local minimizer of ϕ.

We are about to extend these second order optimality conditions to the case of constrained problems. The extension will go in two stages: first, we shall consider the case of a very special constrained problem

$$(P^*) \quad \min_y \left\{ \phi(y) \quad \phi_1(y) \equiv y_1 \leq 0, ..., \phi_q(y) \equiv y_q \leq 0, \right. \\ \left. \phi_{q+1}(y) \equiv y_{q+1} = 0, ..., \phi_{q+k}(y) \equiv y_{q+k} = 0, \right\}$$

where ϕ is smooth function and the solution to be tested for local optimality is $y^* = 0$. Then we easily will pass from this special case to the general one, using Theorem 4.1.1.

Case of Special problem (P^*)**.** We start with the necessary optimality condition, which is immediate:

Proposition 4.2.1 *Let ϕ in (P^*) be twice continuously differentiable in a neighbourhood of $y^* = 0$. If $y^* = 0$ is locally optimal for (P^*), then there exist uniquely defined Lagrange multipliers $\lambda_i^* \geq 0$, $i = 1, ..., q$, and μ_j^*, $j = 1, ..., k$, such that for the Lagrange function*

$$L^*(y; \lambda, \mu) = \phi(y) + \sum_{i=1}^{q} \lambda_i \phi_i(y) + \sum_{j=1}^{k} \mu_j \phi_{q+j}(y)$$

of problem (P^*) one has:

- (i) *The gradient in y of L^* taken at the point $(0; \lambda^*, \mu^*)$ vanishes:*

$$\nabla_y L^*(0; \lambda^*, \mu^*) \equiv \nabla\phi(0) + \sum_{i=1}^{q} \lambda_i^* \nabla\phi_i(0) + \sum_{j=1}^{k} \mu_j^* \nabla\phi_{k+j}(0) = 0 \tag{4.2.1}$$

- (ii) *The Hessian of L^* in y taken at the point $(0; \lambda^*, \mu^*)$ is positive semidefinite on the linear subspace $T^* = \{y \mid y_k = ... = y_{q+k} = 0\}$:*

$$d = (d_1, ..., d_n) \in \mathbf{R}^n, d_1 = ... = d_{q+k} = 0 \Rightarrow d^T \nabla_y^2 L^*(0; \lambda^*, \mu^*) d \geq 0. \tag{4.2.2}$$

Proof is immediate. (i) is the KKT condition (which in our case is necessary for optimality due to Theorem 4.1.2, since $y^* = 0$ clearly is a regular feasible solution to the problem. We can also see directly why (i) is true: a direction d with $d_i \leq 0,$, $i = 1, ..., q$, and $d_i = 0$, $i = q + 1, ..., q + k$ (the remaining entries in d can be arbitrary) clearly is a feasible direction for our problem at the point $y^* = 0$, i.e., $y^* + td = td$ is feasible solution to the problem for all small positive – in fact, for all positive – values of t. Since $y^* = 0$ is locally optimal for (P^*), the objective ϕ locally cannot decrease along such a direction: $d^T \nabla\phi(0) \geq 0$. Indeed, if $d^T \nabla\phi(0)$ were negative, then the values of the objective at the feasible solutions td to (P^*) would be, for small positive t, strictly less than the value of the objective at $y^* = 0$, which is impossible, since $y^* = 0$ is locally optimal. Now, what does it mean that

$$d^T \nabla\phi(x^*) \equiv \sum_{i=1}^{n} d_i \frac{\partial}{\partial y_i} \phi(0) \geq 0$$

for all d with the first q entries being nonpositive, the next k entries being zero and the remaining entries being arbitrary? It clearly means that the first q components of $\nabla\phi(0)$ are nonpositive, the next k components may me arbitrary, and the remaining components are zero. Now let us set $\lambda_i^* = -\frac{\partial}{\partial y_i}\phi(0)$, $i = 1, ..., q$ (note that λ_i^* are nonnegative) and $\mu_j^* = -\frac{\partial}{\partial y_{q+j}}\phi(0)$, $j = 1, ..., k$; then the vector

$$\nabla\phi(0) + \sum_{i=1}^{q} \lambda_i^* \nabla\phi_i(0) + \sum_{j=1}^{k} \mu_j^* \nabla\phi_{q+j}(0)$$

will be 0 (note that $\nabla\phi_i$ are simply the basic orths of $\mathbf{R}^n$), and we get the Lagrange multipliers required in (i); their uniqueness is an immediate consequence of linear independence of $\nabla\phi_i(0)$.

(ii) also is immediate: the entire linear subspace T^* is feasible for (P^*), so that from local optimality of $y^* = 0$ for (P) it follows that 0 is local minimizer of ϕ on the linear subspace T^*. But we know what is necessary optimality condition for this latter phenomenon, since the problem of minimizing ϕ on a *linear subspace* is in equivalent to unconstrained minimization problem: we simply should treat T^*, and not $\mathbf{R}^n$, as our universe. From the above second order necessary optimality condition for unconstrained minimization we conclude that the second order derivative of ϕ taken at the point $y^* = 0$ along any direction $d \in T^*$, i.e., the quantity $d^T \nabla^2 \phi(0) d$, should be nonnegative. But this quantity is the same as $d^T \nabla_y^2 L^*(0; \lambda^*, \mu^*) d$, since $\phi(y)$ differs from $L^*(y; \lambda^*, \mu^*)$ by a linear function of y, and we come to (ii).

One may ask: why should we express a simple thing – nonnegativity of the second order derivative of ϕ taken at $y^*=0$ along a direction from T^* – in that strange way – as similar property of the Lagrange function. The answer is that this is the L^*-form of this fact, not the ϕ-form of it, which is stable with respect to nonlinear substitutions of the argument and is therefore appropriate for extension from the case of special problem (P^*) to the case of general optimization problem (P).

■

Now we know what is the necessary second order optimality condition for our special problem (P^*), and what we are about to do is to understand how the condition should be strengthened in order to become sufficient for local optimality. The second order part of our necessary condition in fact comes from the second order part of the necessary optimality condition for the unconstrained case – we simply replace the entire $\mathbf{R}^n$ by the "unconstrained part of the feasible set" – the linear subspace T^*, and the condition says that $\nabla^2\phi(0)\equiv\nabla_y^2L^*(0;\lambda^*,\mu^*)$ should be positive semidefinite along the directions from T^*. By analogy with the unconstrained case we could guess that to make our necessary condition sufficient, we should replace "positive semidefinite" in the latter sentence by "positive definite". The truth, however, is more sophisticated, as is seen from the following example:

$$\min_y\left\{\phi(y)=y_2^2-y_1^2:\phi_1(y)\equiv y_1\le 0\right\}$$

$(q=1,\ k=0)$. Here the "first-order" part of the optimality condition from Proposition 4.2.1 is satisfied with $\lambda_1^*=0$, the linear subspace T^* is $T^*=\{d=(d_1,d_2)\mid d_1=0\}$ and the Hessian of ϕ taken at 0 and restricted to T^* is positive definite: $\frac{\partial^2}{\partial y_2^2}\phi(0)=2>0$. Nevertheless, $y^*=0$ clearly is not locally optimal, and we see that "naive" modification of the condition – "replace $d^T\nabla^2L^*d\ge 0$ by $d^T\nabla^2L^*d>0$ for nonzero $d\in T^*$" – does not work: it does not make the condition sufficient for local optimality.

The actual sufficient condition for local optimality is as follows:

Proposition 4.2.2 *Consider special optimization program (P^*), and assume that the data ϕ,ϕ_i are twice continuously differentiable in a neighbourhood of $y^*=0$. Assume also that there exist Lagrange multipliers $\lambda_i^*\ge 0$, $i=1,...,q$, and μ_j^*, $j=1,...,k$, such that for the Lagrange function*

$$L^*(y;\lambda,\mu)=\phi(y)+\sum_{i=1}^q\lambda_i\phi_i(y)+\sum_{j=1}^k\mu_j\phi_{q+j}(y)$$

one has

- [first-order part] $\nabla_yL^*(0;\lambda^*,\mu^*)=0$
  and
- [second-order part] $d^T\nabla_y^2L^*(0;\lambda^*,\mu^*)d>0$ *for all nonzero d from the linear subspace*

$$T^{**}=\{d\in\mathbf{R}^n\mid\lambda_i^*d_i=0,\ i=1,...,q,d_{q+1}=...=d_{q+k}=0\}.$$

Then $y^*=0$ is locally optimal solution to (P^*).

Before proving this statement, let us stress the difference between the necessary optimality condition from Proposition 4.2.1 and our new sufficient optimality condition. The first order parts of the conditions are identical. The second order part of the sufficient optimality condition is, as it should be, stronger than the one of the necessary condition, and it is stronger in two points:

– first, now we require positive definiteness of the Hessian $\nabla_y^2L^*$ of the Lagrange function along certain subspace of directions, not simply positive semidefiniteness of it, as it was in the necessary condition for optimality. There is no surprise – this is the case already in the unconstrained situation;

– second, and more important, the subspace of directions along which we require definiteness of the Hessian is wider for the sufficient condition than for the necessary one. Indeed, T^{**} and T^* impose the same requirements on the entries of d with indices $\ge q$, and at the same time impose different requirements

on the entries of d with indices $\le q$: T^* requires all these entries to be zero, while T^{**} asks to be zero *only the entries d_i, $i \le q$, associated with positive Lagrange multipliers λ_i^*; the entries of d associated with zero Lagrange multipliers now could be arbitrary.*
Note that the above "bad example" does not satisfy the second order sufficient condition for optimality (and how it could? In this example $y^* = 0$ is not locally optimal!) exactly due to the second of the above two points: in this example $\lambda_1^* = 0$, which makes T^{**} larger than T^* (T^* is the axis of y_2, T^{**} is the entire plane of y_1, y_2); the Hessian of the Lagrange function in this example is positive definite on T^*, but is not positive definite on T^{**}.

Note also that the second of the above two "strengthening points" is important only in the case when some of inequality constraints $\phi_i(y) \le 0$, $i = 1, ..., q$, are associated with zero Lagrange multipliers λ_i^*; this is the only case when T^{**} may be indeed wider than T^*.
Proof of Proposition 4.2.2 is as follows. Assume that the condition for optimality stated in the Proposition is satisfied; we should prove that $y^* = 0$ is locally optimal solution to (P^*). Assume, on contrary, that it is not the case: there exists a sequence $\{y^t\}$ of feasible solutions to (P^*) which converges to $y^* = 0$ and is such that $\phi(y^t) < \phi(0)$, and let us lead this assumption to a contradiction. Of course, we have $y^t \neq 0$ (since $\phi(y^t) < \phi(0)$). Let s_t be the norm, and

$$d^t = s_t^{-1} y^t$$

be the direction of y^t. Since $y^t \to y^* = 0$ as $t \to \infty$, we have $s_t \to 0$. Further, the unit vectors d^t belong to the unit sphere $\{d \mid |d| = 1\}$ of $\mathbf{R}^n$, which is a compact set; therefore, passing to a subsequence, we may assume that the directions d^t converge, as $t \to \infty$, to certain unit vector d^*.

Now let us write down the second-order Taylor expansion of ϕ at $y^* = 0$:

$$\phi(y) - \phi(0) = p^T y + \frac{1}{2} y^T H y + \rho(y), \tag{4.2.3}$$

where $p = \nabla\phi(0)$, $H = \nabla^2\phi(0)$ and $\rho(y)$ is the remainder:

$$|\rho(y)| \le \theta(|y|) \quad [\theta(s) = o(s^2)].$$

From the first order part of our optimality condition we know that

$$p + \sum_{i=1}^{q} \lambda_i^* \nabla\phi_i(0) + \sum_{j=1}^{k} \mu_j^* \nabla\phi_{q+j}(0) = 0;$$

recalling what are ϕ_i, we conclude that (4.2.3) can be rewritten as

$$\phi(y) - \phi(0) = -\sum_{i=1}^{q} \lambda_i^* y_i - \sum_{j=1}^{k} \mu_j^* y_{q+j} + \frac{1}{2} y^T H y + \rho(y). \tag{4.2.4}$$

Substituting $y = y^t = s_t d^T$ and taking into account that the entries with indices $q+1, ..., q+k$ in d^t vanish (since y^t is feasible for (P^*)), we get

$$0 > s_t^{-1}[\phi(y^t) - \phi(0)] = \left[-\sum_{i=1}^{q} \lambda_i^* d_i^t\right]_1 + s_t \left[\frac{1}{2}(d^t)^T H d^t + s_t^{-2}\rho(s_t d^t)\right]_2. \tag{4.2.5}$$

Now let us make several simple observations:

- (a): The quantities $[\cdot]_1$ in the right hand side of (4.2.5) are nonnegative (since y^t is feasible, we have $d_i^t \le 0$, $i = 1, ..., q$, while λ_i^* are nonnegative);
- (b): As $t \to \infty$, the quantities $[\cdot]_1$ converge to $-\sum_{i=1}^{q} \lambda_i^* d_i^* \ge 0$;
- (c): As $t \to \infty$, the quantities $[\cdot]_2$ converge to $\frac{1}{2}(d^*)^T H d^*$ (since $|\rho(s_t d^t)| \le \theta(s_t)$ and $\theta(s) = o(s^2)$), whence the quantities $s_t[\cdot]_2$ converge to 0.

Our first conclusion from these observations is that the right hand side of (4.2.5) converges, as $t \to \infty$, to $-\sum_{i=1}^{q} \lambda_i^* d_i^* \geq 0$, while the left hand side is nonpositive, whence the limit of the right hand side actually is equal to 0:

$$(!) \quad \sum_{i=1}^{q} \lambda_i^* d_i^* = 0.$$

Since the vector d^* inherits from the vectors d^t the property to have the first q coordinates nonpositive and the next k coordinates zero, we see that every term $\lambda_i^* d_i^*$ in the sum (!) is nonpositive; since the sun is equal to 0, we conclude that $\lambda_i^* d_i^* = 0$, $i = 1, ..., q$. Since, besides this, $d_{q+j}^* = 0$, $j = 1, ..., k$, we conclude that $d \in T^{**}$.

Now let us divide both sides of (4.2.5) by s_t; taking into account observation (a), we shall get the inequality

$$0 \geq s_t^{-2}(\phi(y^t) - \phi(0)) \geq [\cdot]_2.$$

From (c), the right hand side in this inequality converges, as $t \to \infty$, to $\frac{1}{2}(d^*)^T H d^*$, so that this quantity should be nonpositive. At the same time, we already know that $d^* \in T^{**}$ and we from the very beginning know that $|d^*| = 1$; thus, part (ii) of the optimality condition implies that $\frac{1}{2}(d^*)^T H d^* > 0$ (note that $\nabla^2\phi(0) = \nabla_y^2 L^*(0; \lambda^*, \mu^*)$: ϕ_i are linear!); this is the desired contradiction. ∎

From special case to the general one. Now we are in a position where we can easily achieve the second of our targets – to get the second order optimality conditions for the case of a general optimization problem (P).

Theorem 4.2.1 [Second Order Optimality Conditions in Mathematical Programming]
Consider constrained optimization program (P), and assume that x^* is a feasible solution to this problem, and the data $f, g_1, ..., g_m, h_1, ..., h_k$ are twice continuously differentiable in a neighbourhood of x^*. Let also x^* be regular, so that the gradients, taken at x^*, of the active at the point constraints of (P) are linearly independent. Finally, let

$$L(x; \lambda, \mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{k} \mu_j h_j(x)$$

be the Lagrange function of the problem.

(i) [Second Order Necessary Optimality Condition] *If x^* is locally optimal for (P), then there exist and are uniquely defined Lagrange multipliers $\lambda_i^* \geq 0$ and μ_j^* such that*

- *(λ^*, μ^*) certify that x^* is a KKT point of (P):*

$$\begin{array}{rcl} \lambda_i^* g_i(x^*) & = & 0,\ i = 1, ..., m; \\ \nabla_x L(x^*; \lambda^*, \mu^*) & \equiv & \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum_{j=1}^{k} \mu_j^* \nabla h_j(x^*) = 0 \end{array} \tag{4.2.6}$$

- *The Hessian $\nabla_x^2 L(x^*; \lambda^*, \mu^*)$ of the Lagrange function in x is positive semidefinite on the orthogonal complement M^* to the set of gradients of active at x^* constraints:*

$$d \in M^* \equiv \{d \mid d^T \nabla g_i(x^*) = 0,\ i \in I(x^*),\ d^T \nabla h_j(x^*) = 0,\ j = 1, ..., k\} \Rightarrow$$
$$d^T \nabla_x^2 L(x^*; \lambda^*, \mu^*) d \geq 0$$

(here $I(x^*)$ is the set of indices of active at x^* inequality constraints).

(ii) [Second Order Sufficient Optimality Condition] *Assume that there exist Lagrange multipliers $\lambda_i^* \geq 0$, μ_j^* such that*

- *λ^*, μ^* certify that x^* is a KKT point of (P), i.e., (4.2.6) is satisfied*

- *The Hessian $\nabla_x^2 L(x^*;\lambda^*,\mu^*)$ of the Lagrange function in x is positive definite on the orthogonal complement M^{**} to the set of gradients of equality constraints and those active at x^* inequality constraints which are associated with positive Lagrange multipliers λ_i^*:*

$$d \in M^{**} \equiv \{d \mid d^T\nabla g_i(x^*) = 0,\ i \in J(x^*),\ d^T\nabla h_j(x^*) = 0,\ j = 1,...,k\},\ d \neq 0 \Rightarrow d^T\nabla_x^2 L(x^*;\lambda^*,\mu^*)d > 0$$

(here $J(x^*)$ is the set of indices of active at x^* inequality constraints associated with positive Lagrange multipliers λ_i^*).

Then x^* is locally optimal for (P).

Proof. Without loss of generality, we may assume that the inequality constraints active at x^* are the first q inequality constraints. Now let us consider the $q+k$ functions $g_1,...,g_q,h_1,...,h_k$; all these functions are equal to 0 at x^*, and their gradients at the point are linearly independent. According to Theorem 4.1.1, there exists at least twice continuously differentiable in a neighbourhood Y of the point $y^* = 0$ substitution of argument $y \mapsto S(y)$, $S(0) = x^*$, which is a one-to-one mapping of Y onto a neighbourhood X of x^*, has at least twice continuously differentiable in X inverse $S^{-1}(x)$ and "linearizes" the functions $g_1,...,g_q,h_1,...,h_k$:

$$\begin{array}{rcl}\phi_i(y) & \equiv & g_i(S(y)) \equiv y_i,\ i = 1,...,q;\\ \phi_i(y) & \equiv & h_{i-q}(S(y)) \equiv y_i,\ i = q+1,...,q+k.\end{array}$$

Now let

$$\phi(y) = f(S(y))$$

and consider the special problem

$$(P^*) \quad \min_y \{\phi(y) : \phi_i(y) \le 0,\ i = 1,...,q,\ \phi_i(y) = 0,\ i = q+1,...,q+k\}.$$

It is immediately seen that

- (a) the functions ϕ,ϕ_i, $i = 1,...,q+k$, are twice continuously differentiable in a neighbourhood of $y^* = 0$
- (b) x^* is locally optimal solution for (P) if and only if $y^* = 0$ is locally optimal solution for (P^*).

In view of these observations, we can get necessary/sufficient conditions for local optimality of x^* in (P) as direct "translations" of the already known to us optimality conditions for y^* in (P^*). The only thing we should understand what are the entities mentioned in (P^*)-optimality conditions in the "(P)-language". This is easy.

First of all, we can write

$$f(x) = \phi(R(x)),\ g_i(x) = \phi(R(x)),\ i = 1,...,q,\ h_j(x) = \phi_{q+j}(R(x)),\ j = 1,...,k, \tag{4.2.7}$$

where $R(\cdot)$ is $S^{-1}(\cdot)$.

Second, we have

$$S(R(x)) \equiv x$$

in a neighbourhood of x^*; differentiating this identity, we get

$$S'(0)R'(x^*) = I$$

(I is the unit matrix), so that the matrix

$$Q = R'(x^*)$$

is nonsingular.

Third, we can forget about inequality constraints of (P) which are non-active at x^* – these constraints neither influence the property of x^* to be locally optimal for (P), nor participate in the optimality conditions we are going to prove. The latter point should be explained: formally, non-active constraints

do participate in the optimality conditions – they appear in the Lagrange function of (P). But this is nothing but illusion (caused by our desire to write the conditions with $\sum_{i=1}^{m}$ instead of $\sum_{i\in I(x^*)}$): both the necessary and the sufficient conditions in question include complementary slackness $\lambda_i^* g_i(x^*) = 0$, $i = 1, ..., m$, which implies that the Lagrange multipliers associated with non-active g_i are zeros; looking at our optimality conditions, we conclude that in fact the nonactive constraints do not enter them.

After we drop the non-active constraints (we just have seen that it can be done), the Lagrange functions of (P) and (P^*) become linked with each other by the same relation as the data functions of the problems:

$$L(x; \lambda, \mu) = L^*(R(x); \lambda, \mu). \tag{4.2.8}$$

Fourth, and the most important, from (4.2.8) and the rules of differentiating superpositions we have the following relations:

$$\nabla_x L(x^*; \lambda, \mu) = Q^T \nabla_y L^*(y^*; \lambda, \mu) \tag{4.2.9}$$

and

$$v^T \nabla_x^2 L(x^*; \lambda, \mu) v = (Qv)^T \nabla_y^2 L^*(y^*; \lambda, \mu)(Qv) + [\nabla_y L^*(0; \lambda, \mu)]^T R''[v],$$

where v is an arbitrary vector and

$$R''[v] = \frac{d^2}{dt^2} \mid_{t=0} R(x^* + tv)$$

is the second-order derivative of the mapping $R(\cdot)$ taken at x^* along the direction v. It follows that *if λ and μ are such that $\nabla_y L^*(0; \lambda, \mu) = 0$, then the Hessians of L and L^* in the primal variables are linked by the simple relation*

$$\nabla_x^2 L(x^*; \lambda, \mu) = Q^T \nabla_y^2 L^*(0; \lambda, \mu) Q \tag{4.2.10}$$

which does not involve the "curvature" $R''[\cdot]$ of the substitution $x \mapsto R(x)$ which converts (P) into (P^*).

Since in the optimality conditions for (P^*) we focus on Lagrange multipliers which make $\nabla L^*(0; \lambda, \mu)$ equal to zero, the latter observation makes the "translation" of optimality conditions from the (P^*)-language to the (P)-language indeed easy. Let me show how this translation is carried out for the necessary optimality condition; with this example, the reader definitely would be able to translate the sufficient optimality condition by himself.

Thus, let us prove (i). Assume that x^* is locally optimal for (P). Then, according to the above remarks, $y^* = 0$ is locally optimal for (P^*). By Proposition 4.2.1 the latter means that there exist $\lambda_i^* \geq 0$ and μ_j^* such that

$$\begin{array}{lrcl} (\#) & \nabla_y L^*(0; \lambda^*, \mu^*) & = & 0 \\ (\&) & d^T \nabla_y^2 L^*(0; \lambda^*, \mu^*) d & \geq & 0 \ \forall d \in T^* \equiv \{d \mid d_i = 0, \ i = 1, ..., q + k\} \end{array} .$$

Let us verify that the above λ^* and μ^* are exactly the entities required in (i)[2]. First, we should not bother about complementary slackness $\lambda_i^* g_i(x^*) = 0$ – see the "recall" in the latter footnote. Second, we do have the KKT Equation $\nabla_x L(x^*; \lambda^*, \mu^*) = 0$, since we have similar equation $(\#)$ for L^* and we have the chain rule (4.2.9). Thus, we do have the first-order part of (i). To get the second-order part, note that we have similar statement $(\&)$ for L^*, and from $(\#)$ it follows that we have also the "chain rule" (4.2.10), so that we have the inequality

$$d^T \nabla_x^2 L(x^*; \lambda^*, \mu^*) d \geq 0 \quad \forall d : Qd \in T^*. \tag{4.2.11}$$

It remains to understand what does it mean that $Qd \in T^*$. We have $\phi_i(y) \equiv y_i$, $g_i(x) = \phi_i(R(x))$, $i = 1, ..., q$, and $h_j(x) = \phi_{q+j}(R(x))$, which, in simple words, means that the vector-function comprised

[2] recall that we have reduced the situation to the case when all inequality constraints in (P) are active at x^*; otherwise we would be supposed to say: "Let us take λ_i^*'s coming from (P^*) as the required in (i) Lagrange multipliers for the active inequality constraints, and let us set the Lagrange multipliers of nonactive constraints to zeros"

of $g_1, ..., g_q, h_1, ..., h_k$ is nothing but the $(q+k)$-dimensional initial fragment of the n-dimensional vector-function R. Since $Q = R'(x^*)$, the first $q+k$ entries of the vector Qd are exactly the inner products of the taken at x^* gradients of $g_1, ..., g_q, h_1, ..., h_k$ with the direction d; to say that $Qd \in T^*$ is the same as to say that all these inner products are zero (recall that T^* is the subspace of vectors with first $q+k$ entries equal to 0), i.e., to say that $d \in M^*$. Thus, (4.2.11) is exactly the second-order part of (i). □

Remark 4.2.1 Now we understand what for we formulated the second-order parts of the optimality conditions in Propositions 4.2.1, 4.2.2 in terms of the Hessian of the Lagrange function and not the objective (although in the case of special problem (P^*) the Hessian of the Lagrangian is exactly the same as the Hessian of the objective). Only the formulation in terms of the Lagrange function remains invariant under nonlinear substitutions of argument – the tool we used to get the optimality conditions in the general case as an immediate consequences of the same conditions for the simple special case.

Geometry of Second Order Optimality Conditions. In fact Second Order Optimality Conditions say to us very simple things. To see it, we first should understand the geometry of the linear subspace M^* involved into the necessary condition. This is the subspace we get as follows: we take all constraints of the problem which are active at locally optimal solution x^*, linearize them at this point and take the affine set which is given by the system of linear equations "linearizations of the active constraints are equal to 0"; this affine set is exactly $x^* + M^*$. What is this set geometrically? Nothing but the tangent plane to the surface $\mathcal{S}$ where all constraints active at x^* still are active. Thus, directions from M^* are the directions tangent to $\mathcal{S}$ at x^*. If $\mathcal{S}$ is enough regular (as it is under the regularity assumption), moving from x^* "forward" and "backward" along such a direction, we stay "very close" to $\mathcal{S}$ – the distance from $\mathcal{S}$ is infinitesimal of highest order as compared to the step along the direction. It follows that if x^* is locally optimal, then no tangent direction could be direction of decrease of the objective. Indeed, otherwise we could improve x^* by (1) performing a small step along this tangent direction (it would improve the objective by something which is of the same order of magnitude as the step). Of course, this displacement, generally speaking, pushes us out of the feasible set, but we still are very close to it – the distance to the feasible set cannot be larger than the distance to $\mathcal{S}$, and the latter distance is much smaller than the step we did. It follows that if we accompany (1) by "correction" (2) – movement from the point given by (1) to the closest feasible point – we perhaps loose in the value of the objective, but the losses are very small – of the highest order as compared to the length of the step (1). The balance, for small stepsizes, would be in our favour – in (1), we gain something almost proportional to the stepsize, in (2) loose something which is much smaller than the stepsize. But a locally optimal solution cannot be improved by small modification, and we conclude that there exist no input to our mechanism – no tangent direction which is a direction of decrease for the objective. In other words,

the gradient of the objective at a locally optimal solution x^* should be orthogonal to the (tangent plane of the) surface $\mathcal{S}$ of active constraints.

This is partial "geometrical translation" of the first order part of the KKT condition. Indeed, we know what is the tangent plane in question – it is the orthogonal complement to the set of gradients of active constraints; the standard Linear Algebra fact $(X^\perp)^\perp = \mathrm{Lin}(X)$ means that it is the same – to say that certain vector is orthogonal to tangent plane and to say that the vector is a linear combination of the gradients of active constraints, as it is said in the KKT condition. Note that in the case of *equality* constrained problem the KKT condition says nothing else, so that in this case orthogonality of $\nabla f(x^*)$ and $\mathcal{S}$ is complete, not partial, geometrical interpretation of the condition. In the inequality constrained case, the KKT condition provides us with additional information: it says what should be the signs of coefficients at the gradients of inequality constraints in the representation of $\nabla f(x^*)$

(these coefficients should be ≤ 0). This conclusion comes from the fact that at a locally optimal x^* the objective has no right to decrease not only along direction tangent to $\mathcal{S}$, but also along those tangent to the surface given by equality constraints and leading inside the "curved half-spaces" given by active inequality constraints. We see that the KKT condition – the first-order part of the second order necessary optimality condition – is very geometric.

What about the geometry of the second-order part of the condition? A naive guess would be like this:

> "if x^* is locally optimal, we for sure are unable to improve the objective by small displacements along $\mathcal{S}$. *Perhaps* it means that we are unable to improve the objective by small displacements along the tangent to $\mathcal{S}$ plane $x^* + M^*$ as well – this tangent plane locally is so close to $\mathcal{S}$! If our guess is true, it means that x^* is local minimizer of the objective on the tangent plane, and we know what does it mean – the gradient of the objective at x^* should be orthogonal to M^*, and the second order derivative of it along every direction from M^* should be nonnegative".

The above reasoning is absolutely false. The second order part of the "unconstrained" optimality condition – "the second order derivative along any direction should be nonnegative" – comes from analyzing second-order effects caused by small perturbations of x^*, and the tangent plane does not, normally, approximate a curved surface within second-order terms. It follows that the second-order phenomena we met when travelling along a curved surface are not the same as those when travelling along the tangent plane to this surface, so that the conclusions derived from analyzing these "plane" phenomena may have nothing in common with reality. For example, in the optimization problem

$$\min_x \left\{-0.01x_1^2 : x_1^2 + x_2^2 = 1\right\}$$

the point $x^* = (0, 1)$ clearly is locally optimal, in spite of the fact that the second order derivative of the objective taken at x^* along the tangent line $\{x_2 = 1\}$ to the feasible circumference is negative.

It turns out that this is the second order derivative of the Lagrange function (with properly chosen Lagrange multipliers), not the one of the objective, which should be nonnegative along tangent directions to $\mathcal{S}$ at a local optimum, and this is the main moral which can be extracted from the second order optimality conditions.

4.2.1 Nondegenerate solutions and Sensitivity Analysis

It makes sense to introduce a special name for "good" locally optimal solutions to (P):

Definition 4.2.1 *A feasible solution x^* of problem (P) is called nondegenerate, if*

- *x^* is regular for the constraints of the problem (Definition 4.1.1),*
- *x^* satisfies the Sufficient Second Order Optimality condition from Theorem 4.2.1, and*
- *x^* is such that all given by the Second Order Optimality condition Lagrange multipliers λ_i^* for the active at x^* inequality constraints are strictly positive rather than simply nonnegative,*

The third requirement imposed in the definition might look strange – what for is it? Already the first two of the requirements enforce x^* to be locally optimal! In fact, all three requirements make sense – all of them are needed by *Sensitivity Analysis*.

Given problem (P), we can embed it into the $m+k$-parameter family of problems

$$(P_{b,d}) \quad \min_x \left\{ f(x) : h(x) \equiv \begin{pmatrix} h_1(x) \\ \dots \\ h_m(x) \end{pmatrix} = b,\ g(x) \equiv \begin{pmatrix} g_1(x) \\ \dots \\ g_k(x) \end{pmatrix} \le d \right\}; \tag{4.2.12}$$

in this notation, (P) itself is the problem $(\mathrm{P}_{0,0})$.

It turns out that the Lagrange multipliers are responsible for the derivatives of the "local optimal value" with respect to the parameters b and d of the family (4.2.12):

Theorem 4.2.2 *Consider parameterized family of problems (4.2.12), let x^* be a nondegenerate solution to problem $(\mathrm{P}_{0,0})$, and let λ^*, μ^* be the corresponding vector of Lagrange multipliers, so that $(x^*; \lambda^*, \mu^*)$ is a KKT point of the problem $(\mathrm{P}_{0,1})$. Then there exist*

- *small enough neighbourhood X of the point x^* in the space $\mathbf{R}^n$ of the design vectors,*
- *small enough neighbourhood V of the point $(b,d) = (0,0)$ in the space $\mathbf{R}^m \times \mathbf{R}^k$ of the parameters,*
- *continuously differentiable functions*

$$x^*(b,d) : V \to X \subset \mathbf{R}^n; \quad \lambda^*(b,d) : V \to \mathbf{R}^m, \quad \mu^*(b,d) : V \to \mathbf{R}^k$$

$$[x^*(0,0) = x^*, \lambda^*(0,0) = \lambda^*, \mu^*(0,0) = \mu^*]$$

such that

- *whenever (b,d) is close enough to $(0,0)$, namely, whenever $(b,d) \in V$, $x^*(b,d) \in X$ is a nondegenerate locally optimal solution to the "perturbed" problem $(\mathrm{P}_{b,d})$, $\lambda^*(b,d)$, $\mu^*(b,d)$ being the corresponding vectors of Lagrange multipliers; $(x^*(b,d), \lambda^*(b,d), \mu^*(b,d))$ is the only KKT point of problem $(P_{b,d})$ in the set of points (x, λ, μ) with $x \in X$.*
- *for $(b,d) \in V$, $-\lambda_i^*(b,d)$ and $-\mu_j^*(b,d)$ are the derivatives of the "local optimal value"*

$$f^*(b,d) \equiv f(x^*(b,d))$$

with respect to b_i, d_j, respectively:

$$\nabla_b f^*(b,d) = -\lambda^*(b,d); \quad \nabla_d f^*(b,d) = -\mu^*(b,d). \tag{4.2.13}$$

Theorem 4.2.2 can be easily derived from the Implicit Function Theorem in the following standard form (which is slightly different from the form presented in Theorem 4.1.1):

> **Implicit Function Theorem.** *Let U' be a neighbourhood of a point $u^* \in \mathbf{R}^N$, V' be a neighbourhood of a point $v^* \in \mathbf{R}^M$, and let*
>
> $$\Phi(u,v) : U' \times V' \to \mathbf{R}^N$$
>
> *be $s \ge 1$ times continuously differentiable vector-valued function of $u \in U'$ and $v \in V'$ such that*
>
> $$\Phi(u^*, v^*) = 0$$
>
> *and the $N \times N$ matrix*
>
> $$\Phi'_u(u^*, v^*) \equiv \frac{\partial}{\partial u}|_{u=u^*, v=v^*} \Phi(u,v)$$
>
> *is nonsingular.*
>
> *Then there exist*
>
> - *a neighbourhood $U \subset U'$ of u^* in $\mathbf{R}^N$ and a neighbourhood $V \subset V'$ of v^* in $\mathbf{R}^M$,*

- *s times continuously differentiable function*

$$u^*(v) : V \to U$$

– *the implicit function given by the system of equations*

$$\Phi(u, v) = 0$$

such that

- $\Phi(u^*(v), v) \equiv 0$, $v \in V$;
- *for every $v \in V$, the point $u^*(v) \in U$ is the unique in U solution of the equality system $\Phi(u, v) = 0$, u being the unknown in the system.*

Here is the sketch of a derivation of Theorem 4.2.2 from the Implicit Function Theorem:

1^0. Notation. Let x^* be a nondegenerate locally optimal solution to $(P_{0,0})$, and let λ^* and μ^* be the corresponding Lagrange multipliers. Without loss of generality, we may assume that all inequality constraints of the problem are active at x^* – the nonactive constraints locally do not influence the problem, and we can drop them.

2^0. Let us set

$$N = n + m + k; \quad M = m + k.$$

We partition vectors $u \in \mathbf{R}^N$ into parts

$$u = (x, \lambda, \mu)$$

of the sizes n, m, k, respectively, and set

$$u^* = (x^*, \lambda^*, \mu^*).$$

Similarly, we partition vectors $v \in \mathbf{R}^M$ into parts

$$v = (b, d)$$

of the sizes m and k, respectively, and set

$$v^* = (0, 0).$$

Now let us define the mapping

$$\Phi(u, v) = \begin{pmatrix} \nabla f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{j=1}^{k} \mu_j \nabla g_j(x) \\ h(x) - b \\ \mu \times (g(x) - d) \end{pmatrix}$$

where $\mu \times q$ is the *dot product* of k-dimensional vectors μ and q, i.e., the k-dimensional vector given by $[\mu \times q]_j = \mu_j q_j$.

Note that the system of N equations

$$\Phi(u, v) = 0$$

is nothing but the equation part of the KKT system corresponding to the problem $(\mathrm{P}_{b,d})$.

3^0. Let us verify that the introduced data satisfy the premise of the Implicit Function Theorem (where one can set $U' = \mathbf{R}^N$, $V' = \mathbf{R}^M$). Indeed,

$$\Phi(u^*, v^*) = 0$$

since x^* is nondegenerate solution of (P_{v^*})=$(P_{0,0})$, and λ^*, μ^* are the corresponding Lagrange multipliers. All we should prove is that the matrix $\Phi'_u(u^*, v^*)$ is nondegenerate. To this end it suffices to verify that

$$\phi'_u(u^*, v^*) \begin{pmatrix} dx \\ d\lambda \\ d\mu \end{pmatrix} = 0$$

only for zero $dx, d\lambda, d\mu$.

Direct computation demonstrates that the latter equality is nothing but the system of linear equalities

$$\begin{array}{lrcl}
(\alpha) & [\nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 h_i(x^*) + \sum_{j=1}^k \mu_j^* \nabla^2 g_j(x^*)]dx & & \\
& + \sum_{i=1}^m d\lambda_i \nabla h_i(x^*) + \sum_{j=1}^k d\mu_j \nabla g_j(x^*) & = & 0 \\
(\beta) & (dx)^T \nabla h_i(x^*) & = & 0,\ i = 1, ..., m \\
(\gamma) & \mu_j (dx)^T \nabla g_j(x^*) & = & 0,\ j = 1, ..., m
\end{array}$$

(when computing last k rows in the matrix $\Phi'_u(u^*, v^*)$, you should take into account that, by virtue of our convention from 1^0, $g_j(x^*) = 0$ for all j, so that the terms $d\mu_j g_j(x^*)$ which occur when differentiating Φ in fact vanish).

From (β) and (γ) it follows that dx is orthogonal to the gradients of all equality constraints and to the gradients of those inequality constraints for which $\mu_j^* > 0$, all the gradients being taken at x^*. With this in mind, let us take the inner product of both sides of (α) (this is an n-dimensional vector equality) and the vector dx; we will get

$$(dx)^T \nabla_x^2 L(x^*; \lambda^*, \mu^*) dx = 0,$$

which, by definition of a nondegenerate solution and due to the already established orthogonality of dx and the gradients of the equality constraints and the gradients of those inequality constraints with $\mu_j^* > 0$, implies that $dx = 0$. Since $dx = 0$, (α) results in

$$\sum_{i=1}^m d\lambda_i \nabla h_i(x^*) + \sum_{j=1}^k d\mu_j \nabla g_j(x^*),$$

whence $d\lambda = 0$ and $d\mu = 0$ (x^* is regular, so that the set comprised of gradients of all the constraints is linearly independent; here again we use our convention that all the inequality constraints are active at x^*). Thus, $dx = 0, d\lambda = 0, d\mu = 0$, as claimed.

4^0. Since, by 3^0, the premise of the Implicit Function Theorem is satisfied at (u^*, v^*) with $s = 1$, the theorem implies that there exists a neighbourhood V of the point $v^* = (0, 0)$ in the plane of parameters identifying problems (P_v) from our family, a neighbourhood U of the point u^* and once continuously differentiable function

$$u^*(v) \equiv (x^*(b, d), \lambda^*(b, d), \mu^*(b, d)) : V \to U \quad [v = (b, d)]$$

such that

A:

$$\Phi(u^*(v), v) \equiv 0, \ \ v \in V,$$

and for $v \in V$ the point $u^*(v)$ is the unique in U solution to the system of equations

$$\Phi(u, v) = 0.$$

Since x^* is nondegenerate, $\mu^* > 0$; since $u^*(\cdot)$ is continuous, passing, if necessary, to smaller U and V, we may assume that

B: $\mu^*(v) > 0$ when $v \in V$.

Now let X' be a neighbourhood of x^* such that the gradients of the functions $h_i(x), g_j(x)$, $i = 1, ..., m$, $j = 1, ..., k$ are linearly independent for $x \in X'$; since these gradients are linearly independent at $x = x^*$ (x^* is a regular point of the constraints!), such a neighbourhood exists. Some points x from X' can be extended by properly chosen Lagrange multipliers to triples (x, λ, μ) satisfying the KKT equation, and some, perhaps, cannot; let X'' be the set of those $x \in X'$ where these multipliers exist. Since the gradients of all the constraints at the points from X' are linearly independent, the Lagrange multipliers for $x \in X''$ are uniquely defined and continuous on X''; when $x = x^*$, these multipliers are λ^*, μ^*. Consequently, there exists small enough neighbourhood, let it be called X, of x^* such that

C: the set of gradients of all the constraints at any point $x \in X$ is linearly independent, and if $x \in X$ can be extended by Lagrange multipliers λ, μ to a triple (x, λ, μ) satisfying the KKT equation, then $(x, \lambda, \mu) \in U$.

Shrinking, if necessary, V, we can ensure that

D: $x^*(v) \in X$ when $v \in V$.

Now note that the matrix $\nabla_x^2 L(x; \lambda, \mu)$ is continuous in x, λ, μ, and the plane

$$\bar{T}(x) = \{e \mid e^T \nabla h_i(x) = 0, e^T \nabla g_j(x) = 0, i = 1, ..., m, j = 1, ..., n\}$$

is, in the natural sense, continuous in $x \in X'$. When $x = x^*$, $\lambda = \lambda^*$, $\mu = \mu^*$, the matrix $\nabla_x^2 L(x; \lambda, \mu)$ is positive definite on the plane $\bar{T}(x^*)$; by continuity reasons, it follows that the matrix $\nabla_x^2 L(x^*(v); \lambda^*(v), \mu^*(v))$ is positive definite on $\bar{T}(x^*(v))$, whenever v is close enough to v^*. Again shrinking V, we may assume that the latter property is satisfied when $v \in V$:

E: the matrix $\nabla_x^2 L(x^*(v); \lambda^*(v), \mu^*(v))$ is positive definite on the linear subspace $\bar{T}(x^*(v))$ when $v \in V$.

Now we are basically done. By construction, when $v = (b, d) \in V$, the point $x^*(v)$ along with Lagrange multipliers $\lambda^*(v), \mu^*(v)$ satisfies the equations of the KKT system (by **A**) and $\mu^*(v) > 0$ (by **B**); the latter two properties clearly imply that $x^*(v)$ is the KKT point for the problem (P_v). By **C** and **E** this point is nondegenerate locally optimal solution to (P_v). Besides this, when $v \in V$, $x^*(v)$ is the only point in X which satisfies the necessary First Order Optimality condition for (P_v). Indeed, let $x \in X$ admit extension, by Lagrange multipliers λ, μ, to a triple satisfying the KKT equation associated with (P_v). By **C** we have $u = (x, \lambda, \mu) \in U$, so that u is a belonging to U solution to the system of equations

$$\Phi(\cdot, v) = 0.$$

By **A** such a solution is uniquely defined and is $(x^*(v), \lambda^*(v), \mu^*(v))$, as claimed.

Thus, we have proved all required statements except for (4.2.13). The proof of the latter relation is immediate: we know from the Implicit Function Theorem that $x^*(v)$ is differentiable in $v \in V$; let x' be the derivative of this mapping, taken at certain point $\bar{v} \in V$ along certain direction $\delta v = (\delta b, \delta d)$. Taking inner product of both sides of the equality

$$\nabla_x f(x^*(\bar{v})) + \sum_{i=1}^{m} \lambda_i^*(\bar{v}) \nabla_x h_i(x^*(\bar{v})) + \sum_{j=1}^{k} \mu^*(\bar{v}) \nabla_x g_j(x^*(\bar{v})) = 0$$

and x' and taking into account that

$$(x')^T \nabla_x f(x^*(\bar{v})) = \frac{d}{dt}|_{t=0} f^*(\bar{v} + t\delta v)$$

("chain rule"), we get

$$\frac{d}{dt}|_{t=0} f^*(\bar{v} + t\delta v) = -\sum_{i=1}^{m} \lambda_i^*(\bar{v})(x')^T \nabla_x h_i(x^*(\bar{v})) - \sum_{j=1}^{k} \mu_j^*(\bar{v})(x')^T \nabla_x g_j(x^*(\bar{v})). \tag{4.2.14}$$

We have, identically in $v = (b, d) \in V$,

$$h_i(x^*(v)) = b_i,\ i = 1, ..., m; \quad g_j(x^*(v)) = d_j,\ j = 1, ..., k$$

(the fact that all g_j are active at $x^*(v)$ comes from the KKT system due to $\mu_j^*(v) > 0$, $j = 1, ..., k$, see **B**. Differentiating these equalities at the point $\bar{v}$ in the direction δv, we observe that the right hand side in (4.2.14) is equal to

$$-\sum_{i=1}^{k} \lambda_i^*(\bar{v}) \delta b_i - \sum_{j=1}^{k} \mu_j^*(\bar{v}) \delta d_j,$$

and (4.2.14) results in (4.2.13).

4.3 Concluding Remarks

It was announced in the preface to the course and to this lecture that optimality conditions in some cases allow to find closed form solutions to optimization problems. After we know what are the optimality conditions, it is time to explain how one can use them to solve a problem "on the paper". The scheme is very simple. Given a constrained optimization problem (P), we can write down the KKT optimality conditions along with the feasibility requirements:

$$\begin{array}{rcll} \nabla f(x^*) + \sum\limits_{i=1}^{m} \lambda_i^* \nabla g_i(x^*) + \sum\limits_{j=1}^{k} \mu_j^* \nabla h_j(x^*) & = & 0 & [n = \dim x \text{ equations}] \\ \lambda_i^* g_i(x^*) & = & 0,\ i = 1, ..., .m & [m \text{ equations}] \\ h_j(x^*) & = & 0,\ j = 1, ..., k & [k \text{ equations}] \\ g_i(x^*) & \leq & 0,\ i = 1, ..., m & \\ \lambda_i^* & \geq & 0,\ i = 1, ..., m & \end{array}$$

The equality part of this system is a system of $n + m + k$ nonlinear equations with $n + m + k$ unknowns – the entries of x^*, λ^*, μ^*. Normally such a system has only finitely many solutions. If we are clever enough to find all these solutions and if by some reason we know that the optimal solution exists and indeed satisfies the KKT condition (e.g., the assumptions of Theorem 4.1.2 are satisfied at every feasible solution), then we may be sure that looking through all solutions to the KKT system and choosing among them the one which is feasible and has the best value of the objective, we may be sure that we shall end up with the optimal solution to the problem. In this process, we may use the inequality part of the system (same as the additional inequalities coming from the second order necessary optimality conditions) to eliminate from the list candidates which do not satisfy the inequalities, which enables to skip more detailed analysis of these candidates.

Approach of this type is especially fruitful if (P) is convex (i.e., $f, g_1, ..., g_m$ are convex and $h_1, ..., h_k$ are linear). The reason is that in this case the KKT conditions are sufficient for global optimality (we know it from the previous Lecture, although for the case when there are no equality constraints; the extension to the case of linear equality constraints is quite straightforward). Thus, if the problem is convex and we are able to point out a single solution to the KKT system, then we can be sure that it is the actual – globally optimal – solution to (P), and we should not bother to look for other KKT points and compare them to each other.

Unfortunately, the outlined program can be carried out in simple cases only; normally nonlinear KKT system is too difficult for analytical solution. Let us look at one – and a very instructive one – of "simple cases".

Minimizing a homogeneous quadratic form over the unit sphere. Consider the problem

$$(\mathrm{Q}) \quad \min_x \left\{ f(x) \equiv x^T A x : g_1(x) \equiv x^T x - 1 \leq 0 \right\},$$

A being a symmetric $n \times n$ matrix. Let us list *all* locally optimal solutions to the problem.

Step 0. Let f^* denote the optimal value. Since $x = 0$ clearly is feasible solution and $f(0) = 0$, we have $f^* \leq 0$. There are, consequently, two possible cases:

Case (A): $f^* = 0$;

Case (B): $f^* < 0$.

Step 1: Case (A). Case (A) takes place if and only if $x^T A x \geq 0$ for all x, $|x| \leq 1$, or, which is the same due to homogeneity with respect to x, if and only if

$$x^T A x \geq 0 \ \forall x.$$

We know that symmetric matrices with this property have special name – they are called *symmetric positive semidefinite* (we met with these matrices in this Lecture and also in the Convexity criterion for twice continuously differentiable functions: such a function is convex in certain open convex domain if and only if the Hessian of the function at any point of the domain is positive semidefinite). In Linear Algebra

there are tests for positive semidefiniteness (the simplest is the Sylvester rule: a symmetric matrix is positive semidefinite if and only if all its principal minors – those formed by several rows and the columns with the same indices as rows – are nonnegative; there are also tests which can be run in cubic in the size of the matrix number of arithmetic operations). Now, what are the locally optimal solutions to the problem in the case of positive semidefinite A? We claim that these are exactly the points x from the unit ball (the feasible set of the problem) which belong to the kernel of A, i.e., are such that

$$Ax = 0.$$

First of all, if x possesses the latter property, then $x^TAx = 0 = f^*$, so that x^* is even globally optimal. Vice versa, assume that x is locally optimal, and let us prove that $Ax = 0$. The constraint in our problem is convex; the objective also is convex (recall the criterion of convexity for smooth functions and note that $f''(x) = 2A$), so that a locally optimal solution is in fact optimal. Thus, x is locally optimal if and only if $x^TAx = 0$. In particular, if x is locally optimal, then, say, $x' = x/2$ also is. At this new locally optimal solution, the constraint is satisfied as a strict inequality, so that x' is an unconstrained local minimizer of function $f(\cdot)$, and by the Fermat rule we get $\nabla f(x') \equiv 2Ax' = 0$, whence also $Ax = 0$, as claimed.

Step 2: Case (B). Now consider the case of $f^* < 0$, i.e., the one when there exists h, $|h| \leq 1$, such that

$$(\#) \qquad h^TAh < 0.$$

What are the locally optimal solutions x^* to the problem?

What is said by the First Order Optimality conditions. Logically, two possibilities can take place: the first is that $|x^*| < 1$, and the second is that $|x^*| = 1$.

Let us prove that the first possibility if in fact impossible. Indeed, in the case of $|x^*| < 1$ x^* should be locally optimal in the *unconstrained* problem $\min\limits_x \{f(x) : x \in \mathbf{R}^n\}$ with smooth objective. By the second order necessary condition for unconstrained local optimality, the Hessian of f at x^* (which is equal to $2A$) should be positive semidefinite, which contradicts (#).

Thus, in the case in question a locally optimal solution x^* is on the boundary of the unit ball, and the constraint $g_1(x) \leq 0$ is active at x^*. The gradient $2x^*$ of this constraint is therefore nonzero at x^*, so that (by Theorem 4.1.2) x^* is a KKT point:

$$\exists \lambda_1^* \geq 0 : \quad \nabla f(x^*) + \lambda_1^* \nabla g_1(x^*) = 0,$$

or, which is the same,

$$Ax^* = -\lambda_1^* x^*.$$

Thus, x^* should be a unit eigenvector[3] of A with a nonpositive eigenvalue $\lambda \equiv -\lambda_1^*$; this is all which we can extract from the First Order Necessary Optimality conditions.

Looking at the example

$$A = \mathrm{Diag}(1, 0, -1, -2, -3, ..., -8)$$

in $\mathbf{R}^{10}$, we see that the First Order Necessary optimality conditions are satisfied by 18 vectors $\pm e_2, \pm e_3, ..., \pm e_{10}$, where e_1, $i = 1, ..., 10$, are the standard basic orths in $\mathbf{R}^{10}$. All these 18 vectors are Karush-Kuhn-Tucker points of the problem, and the First Order Optimality conditions do not allow to find out which of these 18 candidates are locally optimal and which are not. To get the answer, we should use the Second Order Optimality conditions.

What is said by the Second Order Optimality conditions. We come back to our general problem (Q); recall that we are in the case of (B). We already know that a locally optimal solution to (Q) is a unit vector, and the set of constraints active at x^* is comprised of our only inequality constraint; its gradient $2x^*$ at x^* is nonzero, so that we have (Regularity), and consequently have the Second Order Necessary Optimality condition (Theorem 4.2.1.(i)). Thus, there should exist Lagrange multiplier $\lambda_1^* \equiv -\lambda \geq 0$ such that

$$2Ax^* - 2\lambda x^* \equiv \nabla f(x^*) + \lambda_1^* \nabla g_1(x^*) = 0$$

[3] recall that an eigenvector of a square matrix M is a nonzero vector e such that $Me = sx$ for certain real s (this real is called the eigenvalue of M associated with the eigenvector e.

and

$$d^T[2A - 2\lambda I]d \equiv d^T[\nabla^2 f(x^*) + \lambda_1^* \nabla^2 g_1(x^*)]d \geq 0$$

(I is the unit matrix) for any d satisfying the condition

$$d^T \nabla g_1(x^*) \equiv 2d^T x^* = 0.$$

In other words, at x^* we should have, for certain $\lambda \leq 0$,

$$Ax^* = \lambda x^* \tag{4.3.1}$$

and

$$d^T(A - \lambda I)d \geq 0 \tag{4.3.2}$$

for all d such that

$$d^T x^* = 0.$$

Let us prove that in fact (4.3.2) is valid *for all* $d \in \mathbf{R}^n$. Indeed, given a $d \in \mathbf{R}^n$, we can decompose it as

$$d = d_1 + d_2,$$

where d_1 is the orthogonal projection of d onto the one-dimensional subspace spanned by x^*, and d_2 is the orthogonal projection of d onto the orthogonal complement to x^*:

$$d_1 = \alpha x^*,\ d_2^T x^* = 0.$$

We have

$$d^T(A - \lambda I)d = (d_1 + d_2)^T(A - \lambda I)(d_1 + d_2) = d_1^T(A - \lambda I)d_1 + 2d_2^T(A - \lambda I)d_1 + d_2^T(A - \lambda I)d_2;$$

from $d_1 = \alpha x_1$ and from (4.3.1) it follows that the first two terms – those containing d_1 – in the last expression are zero, and from (4.3.2) it follows that the third term is nonnegative (recall that $d_2^T x^* = 0$). Consequently, the expression is nonnegative, as claimed.

We conclude that a locally optimal solution x^* of the problem (Q) should be a unit vector such that x^* is an eigenvector of A with a nonpositive eigenvalue λ and, besides this, such that the matrix $A - \lambda I$ should be positive semidefinite – (4.3.2) should be valid for all d. It follows that λ is the *smallest* of eigenvalues of A: indeed, if λ' is another eigenvalue, so that there exists a nonzero x' with $Ax' = \lambda' x'$, then we should have

$$0 \leq (x')^T(A - \lambda I)x' \equiv (x')^T(\lambda' - \lambda)x' = (\lambda' - \lambda)|x'|^2,$$

whence $\lambda' \geq \lambda$.

We conclude that the λ - the minus Lagrange multiplier associated with a locally optimal solution x^* is *uniquely defined by the problem*: this is the smallest eigenvalue of A. And since

$$f(x^*) = (x^*)^T A x^* = \lambda |x^*|^2 = \lambda$$

(we already know that a locally optimal solution must be a unit eigenvector of A with the eigenvalue λ), we conclude that the value of the objective at a locally optimal solution also is uniquely defined by the problem. Since the problem clearly is solvable (the feasible set is compact and the objective is continuous), locally optimal solutions exist and among them there are optimal ones; and since the objective, as we have seen, is constant on the set of locally optimal solutions, we conclude that

In the case of (B) locally optimal solutions are the same as optimal ones and all of them are unit eigenvectors of A associated with the smallest eigenvalue of the matrix.

On the other hand, if x^* is a unit eigenvector of A with the eigenvalue λ, then $f(x^*) = \lambda$ (see the above computation), so that *all* unit eigenvectors of A associated with the smallest eigenvalue of the matrix are, in the case of (B), optimal solutions to (Q). We see that the Second Order Necessary Optimality condition, in contrast to the First Order one, allows us to get complete description of the solutions to (Q).

Remark 4.3.1 A byproduct of our reasoning is the statement that *if a symmetric matrix A satisfies (#), then there exists an eigenvector of A* ((Q) for sure is solvable, and the First Order Necessary condition says, as we have seen, that an optimal solution must be an eigenvector). Note that it is far from being clear in advance why a symmetric matrix should have an eigenvector. Of course, our reasoning establishes the existence of an eigenvector only under assumption (#), but this restriction can be immediately eliminated (given an a arbitrary symmetric matrix A', one can apply the reasoning to the matrix $A = A' - TI$ which, for large T, for sure satisfies (#), and to get existence of an eigenvector of A; of course, it will be also an eigenvector of A').

The existence of an eigenvector for a symmetric matrix is, of course, a perfectly well known elementary fact of Linear Algebra; here is a several-line proof:

> Let us prove first that an arbitrary matrix A, even with complex entries, possesses a complex eigenvalue. Indeed, λ is an eigenvalue of A if and only if there exists a nonzero (complex) vector z such that $(A-\lambda I)z = 0$, i.e., if and only if the matrix $\lambda I - A$ is singular, or, which is the same, the determinant of the matrix is zero. On the other hand, the determinant of the matrix $\lambda I - A$ clearly is a nonconstant polynomial of λ, and such a polynomial, according to FTA – the Fundamental Theorem of Algebra – has a root; such a root is an eigenvalue of A.
> Now we should prove that if A is real and symmetric, then it has a real eigenvalue and a real eigenvector. This is immediate: we simply shall prove that all eigenvalues of A are real. Indeed, if λ is an eigenvalue of A (regarded as a complex matrix) and z is the corresponding (complex) eigenvector, then the expression
>
> $$\sum_{i,j=1}^{n} A_{ij} z_j z_i^*$$
>
> (* means complex conjugation) is real (look at its conjugate!); on the other hand, for the eigenvector z we have $\sum\limits_j A_{ij} z_j = \lambda z_i$, so that our expression is $\lambda \sum\limits_{i=1}^{n} z_i z_i^* = \lambda \sum\limits_{i=1}^{n} |z_i|^2$; since $z \neq 0$, this latter expression can be real if and only if λ is real.
> Finally, after we know that an eigenvalue λ of a real symmetric matrix (regarded as a matrix with complex entries) in fact is real, we can immediately prove that the eigenvector associated with this eigenvalue also can be chosen to be real: indeed, the real matrix $\lambda I - A$ is singular and has therefore a nontrivial kernel.

In fact all results of our analysis of (Q) can be immediately derived from several other basic facts of Linear Algebra (namely, from possibility to represent a quadratic form $x^T A x$ in a diagonal form $\sum\limits_{i=1}^{n} \lambda_i u_i^2(x)$, $u_i(x)$ being the coordinates of x in a properly chosen orthonormal basis). Thus, in fact in our particular example the Optimization Theory with its Optimality conditions is, in a sense, redundant. Two things, however, should be mentioned:

- The Linear Algebra proof of the existence of an eigenvector is based on the FTA which states existence of a (complex) root of a polynomial. To get the same result on the existence of an eigenvector, in our proof (and in all the proofs it is based upon) we *never* used something like FTA! All we used from Algebra was the elementary theory of systems of linear equations, and we never thought about complex numbers, roots of polynomials, etc.!

  This is an example of what is Mathematics, and it would be very useful exercise for a mathematician to trace back both theories to see what are the "common roots" of two quite different ways to prove the same fact[4)].

[4)] the only solution to this exercise which comes to my mind is as follows: *the simplest proof of the Fundamental Theorem of Algebra is of optimization nature*. This hardly is the complete answer, since the initial proof of FTA given by Gauss is *not* of optimization nature; the only fact from Analysis used in this proof is that a continuous function on the axis which takes both positive and negative values has a zero (Gauss used this fact for a polynomial of an odd degree). This is too far relationship, we think.

- It is worthy of mentioning that the Optimization Theory (which seems to be redundant to establish the existence of an eigenvector of a symmetric matrix) becomes unavoidable when proving a fundamental infinite-dimensional generalization of this fact: the theorem (Hilbert) that *a compact symmetric linear operator in a Hilbert space possesses an eigenvector* [and, finally, even an orthonormal basis comprised of eigenvectors]. We are not going to explain what all these words mean; roughly speaking, it is said that a *infinite dimensional* symmetric matrix can be diagonalized in a properly chosen orthonormal basis (e.g., an integral operator $f(s) \mapsto \int_0^1 K(t,s)f(s)ds$ with not that bad (e.g., square summable) symmetric ($K(t,s) = K^*(s,t)$) kernel K possesses a complete in $L_2[0,1]$ orthonormal system of eigenfunctions; this fact, in particular, explains why the atomic spectra are discrete rather than continuous). When proving this extremely important theorem, one cannot use Linear Algebra tools (there are no determinants and polynomials anymore), but still can use the optimization ones (compactness of the operator implies solvability of the corresponding problem (Q), and the first order necessary optimality condition which in the case in question says that the solution is an eigenvector of the operator, in contrast to FTA, is "dimension-invariant" and remain valid in the infinite-dimensional case as well).

Lecture 5

Optimization Methods: Introduction

This lecture starts the second part of our course; what we are interested in from now on are *numerical methods for nonlinear continuous optimization*, i.e., for solving problems of the type

$$\textit{minimize} \;\; f(x) \;\; s.t. \;\; g_i(x) \leq 0,\, i = 1, ..., m;\; h_j(x) = 0,\, j = 1, ..., k. \tag{5.0.1}$$

Here x varies over $\mathbf{R}^n$, and the *objective* $f(x)$, same as the functions g_i and h_j, are smooth enough (normally we assume them to be at least once continuous differentiable). The constraints

$$g_i(x) \leq 0,\, i = 1, ..., m; \quad h_j(x) = 0,\, j = 1, ..., k$$

are referred to as *functional* constraints, divided in the evident manner into *inequality* and *equality* constraints.

We refer to (5.0.1) as to *nonlinear* optimization problems in order to distinguish between these problems and Linear Programming programs; the latter correspond to the case when all the functions f, g_i, h_j are linear. And we mention *continuous optimization* in the description of our subject to distinguish between it and *discrete optimization*, where we look for a solution from a discrete set, e.g., comprised of vectors with integer coordinates (Integer Programming), vectors with 0-1 coordinates (Boolean Programming), etc.

Problems (5.0.1) arise in huge variety of applications, since whenever people make decisions, they try to make them in an "optimal" manner. If the situation is simple enough, so that the candidate decisions can be parameterized by finite-dimensional vectors, and the quality of these decisions can be characterized by finite set of "computable" criteria, the concept of "optimal" decision typically takes the form of problem (5.0.1). Note that in real-world applications this preliminary phase – *modelling the actual decision problem as an optimization problem with computable objective and constraints* – is, normally, much more difficult and creative than the subsequent phase when we solve the resulting problem. In our course, anyhow, we do not touch this modelling phase, and focus on technique for solving optimization programs.

Recall that we have developed optimality conditions for problems (5.0.1) in Lecture 7. We remember that one can form a square system of nonlinear equations and a system of inequalities which together define certain set – the one of *Karush-Kuhn-Tucker* points of the problem – which, under mild regularity conditions, contains all optimal solutions to the problem. The Karush-Kuhn-Tucker system of equations and inequalities typically has finitely many solutions, and if we are clever enough to find all of them analytically, then we could look through them and to choose the one with the best value of the objective, thus getting the optimal solution in a closed analytical form. The difficulty, however, is that as a rule we are *not* so clever to solve analytically the Karush-Kuhn-Tucker system, same as are unable to find the optimal solution analytically by other means. In all these "difficult" cases – and basically all optimization problems coming from real world applications are difficult in this sense – all we may hope for is a numerical routine, an algorithm which allows to approximate numerically the solutions we are interested in. Thus, numerical optimization methods form the main tool for solving real-world optimization problems.

5.1 Preliminaries on Optimization Methods

It should be stressed that one hardly can hope to design a single optimization method capable to solve efficiently all nonlinear optimization problems – these problems are too diverse. In fact there are numerous methods, and each of them is oriented onto certain restricted family of optimization problems.

5.1.1 Classification of Nonlinear Optimization Problems and Methods

Traditionally, the Nonlinear Optimization Problems (5.0.1) are divided into two large classes:

- *Unconstrained problems* – no inequality or equality constraints are present. The generic form of an unconstrained problem, consequently, is

$$\textit{minimize} \quad f(x) \quad s.t. \quad x \in \mathbf{R}^n, \tag{5.1.1}$$

  f being smooth (at least once continuously differentiable) function on $\mathbf{R}^n$;

- *Constrained problems*, involving at least one inequality or equality constraint.

The constrained problems, in turn, are subdivided into several classes, according to whether there are nonlinear constraints, inequality constraints, and so on; in the mean time we shall speak about this in more details.

According to the outlined classification of optimization problems, the optimization methods are primarily partitioned into those for unconstrained and constrained optimization. Although the more simpler unconstrained problems are not that frequently met in applications, methods for unconstrained optimization play very important role: they are used directly to solve unconstrained problems and indirectly, as building blocks, in many methods of constrained minimization.

5.1.2 Iterative nature of optimization methods

Methods for numerical solving nonlinear optimization problems are, in their essence, *iterative routines*: as applied to problem (5.0.1), a method typically is unable to find exact solution in finite number of computations. What a method generates, is an infinite sequence $\{x_t\}$ of *approximate solutions*. The next *iterate* x_{t+1} is formed, according to certain rules, on the basis of *local information* of the problem collected along the previous iterates. The portion of information $\mathcal{I}_t$ obtained at a current iterate x_t is a vector comprised of the values of the objective and the constraints at x_t and, possibly, of the gradients or even higher-order derivatives of these functions at x_t. Thus, when forming x_{t+1}, the method "knows" the values and the derivatives, up to certain fixed order, of the objective and the constraints at the previous iterates $x_1, ..., x_t$, and this information is all information on the problem available to the method when it generates the new iterate x_{t+1}. This iterate, consequently, is certain function of the information accumulated so far:

$$x_{t+1} = X_{t+1}(\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_t).$$

The collection of the *search rules* $X_t(\cdot)$ predetermines behaviour of the method as applied to an arbitrary problem; consequently, the *method itself can be identified with the collection* $\{X_t\}_{t=1}^{\infty}$. Note that the list of arguments in X_t is comprised of $(t-1)$ portions of local information; in particular, the list of arguments in the very first search rule X_1 is empty, so that this "function" is simply a fixed vector specified by the description of the method – the *starting point*.

From the outlined general scheme of an iterative routine it follows that optimization methods are classified not only according to the types of problems the methods are solving, but also according to the type of local information they use. From this "information" viewpoint, the methods are divided into

- *zero-order* routines using only values of the objective and the constraints and not using their derivatives;

- *first-order* routines using the values and the gradients of the objective and the constraints;

- *second-order* routines using the values, the gradients and the Hessians (i.e., matrices of second-order derivatives) of the objective and the constraints.

In principle, of course, we could speak about methods of orders higher than 2; these methods, however, are never used in practice. Indeed, to use a method of an order k, you should provide possibility to compute partial derivatives of the objective and the constraints up to order k. In the multi-dimensional case it is not that easy even for $k = 1$ and even in the case when your functions are given by explicit analytical expressions (which is not always the case). And there is "explosion of difficulties" in computing higher order derivatives: for a function of n variables, there are n first order derivatives to be computed, $\frac{n(n+1)}{2}$ second order derivatives, $\frac{n(n+1)(n+2)}{2\times 3}$ third order derivatives, etc.; as a result, even in the medium-scale case with n being several tens the difficulties with programming, computation time, memory required to deal with higher-order derivatives make exploiting these derivatives too expensive. On the other hand, there are no serious theoretical advantages in methods of order higher than 2, so there is no compensation for the effort of computing these derivatives.

5.1.3 Convergence of Optimization Methods

As a matter of fact, we cannot expect a nonlinear problem to be solved *exactly* in finite number of steps; all we can hope for is that the sequence of iterates $\{x_t\}$ generated by the method in question converges, as $t \to \infty$, to the solution set of the problem. In the theory of Numerical Optimization, the convergence of an optimization method on certain family of problems is exactly what gives the method right to be qualified as a tool for solving problems from the family. Convergence is not the only characteristic of a method, but this is *the* property which makes a method theoretically valid optimization routine.

5.1.3.1 Rates of convergence

The fact of convergence of an optimization (and any other) computational method is the "weakest" property which gives the method right to exist. In principle, there are as many methods with this property as you wish, and the question is how to rank these methods and which of them to recommend for practical usage. In the traditional Nonlinear Optimization this problem is resolved by looking at the *asymptotic rate of convergence* measured as follows.

> Assume that the method as applied to problem P generates sequence of iterates converging to the solution set X_P^* of the problem. To define the *rate of convergence*, we first introduce an *error function* $\mathrm{err}(x)$ which measures the quality of an approximate solution x; this function should be positive outside X_P^* and should be zero at the latter set.
>
> There are several reasonable choices of the error function. For example, we always can use as it the distance from the approximate solution to the solution set:
>
> $$\mathrm{dist}_P(x) = \inf_{x^* \in X_P^*} |x - x^*|;$$
>
> another choice of the error function could be the residual in terms of the objective and constraints, like
>
> $$\mathrm{res}_P(x) = \max\{f(x) - f^*; [g_1(x)]_+; ...; [g_m(x)]_+; |h_1(x)|; ...; |h_k(x)|\},$$
>
> f^* being the optimal value in P and $[a]_+ = \max(a, 0)$ being the positive part of a real a, etc.
>
> For properly chosen error function (e.g., for dist_P), convergence of the iterates to the solution set implies that the scalar sequence
>
> $$r_t = \mathrm{err}(x_t)$$
>
> converges to 0, and we measure the quality of convergence by the rate at which the nonnegative reals r_t tend to zero.

The standard classification here is as follows:

- [linear convergence] a sequence $\{r_t \geq 0\}$ such that for some $q \in (0,1)$, some $C < \infty$ and all t one has
$$r_t \leq Cq^t$$
is called *linearly converging to 0 with ratio* q; the simplest example of such a sequence is $r_t = Cq^t$. The lower bound of those q for which $\{r_t\}$ linearly converges to 0 with the convergence ratio q is called the *convergence ratio* of the sequence.

  For example, for the sequence $r_t = Cq^t$, same as for the sequence $\{r_t = C(q+\epsilon_t)^t\}$, $\epsilon_t \to 0$, $t \to \infty$, the convergence ratio is q, although the second sequence, generally speaking, does not converge to 0 with the ratio q (it, anyhow, converges to 0 linearly with convergence ratio q' for any $q' \in (q,1)$).

  It is immediately seen that a sufficient condition for a sequence $\{r_t > 0\}$ to be linearly converging with ratio $q \in (0,1)$ is to satisfy the property
$$\limsup_{t\to\infty} \frac{r_{t+1}}{r_t} < q.$$

- [sub- and superlinear convergence] a sequence which converges to 0, but is not linearly converging (e.g., the sequence $r_t = t^{-1}$), is called *sublinearly* converging. A sequence which linearly converges to zero with any positive ratio (so that the convergence ratio of the sequence is 0) is called *superlinearly converging* (e.g., the sequence $r_t = t^{-t}$).

  A sufficient condition for a sequence $\{r_t > 0\}$ to be superlinearly converging is
$$\lim_{t\to\infty} \frac{r_{t+1}}{r_t} = 0.$$

- [convergence of order $p > 1$] a sequence $\{r_t \geq 0\}$ converging to 0 is called to have *convergence order* $p > 1$, if for some C and all large enough t one has
$$r_{t+1} \leq Cr_t^p.$$

  The upper bound of those p for which the sequence converges to 0 with order p is called the *order of convergence* of the sequence.

  For example, the sequence $r_t = a^{(p^t)}$ $(a \in (0,1), p > 1)$ converges to zero with order p, since $r_{t+1}/r_t^p = 1$. The sequences converging to 0 with order 2 have special name – they are called *quadratically convergent*.

  Of course, a sequence converging to 0 with order $p > 1$ is superlinearly converging to 0 (but, generally speaking, not vice versa).

Traditionally, the rate of convergence of iterative numerical routines is measured by the rank of the corresponding sequence of errors $\{r_t = \mathrm{err}(x_t)\}$ in the above scale; in particular, we may speak about *sublinearly*, *linearly*, *superlinearly*, *quadratically* converging methods, same as about methods with order of convergence $p > 1$. It is common to think that the better is the rate of convergence of a method, the more preferable is the method itself. Thus, a linearly converging method is thought to be better than a sublinearly converging one; among two linearly converging methods the more preferable is the one with the smaller convergence ratio; a superlinearly converging method is preferred to a linearly converging one, etc. Of course, all these preferences are "conditional", provided that there are no significant differences in computational complexity of steps, etc.

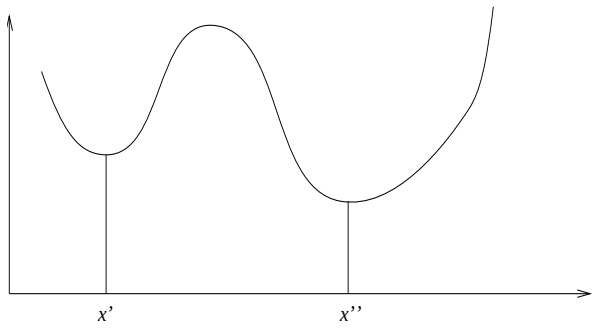
We should stress that the rate of convergence, same as the very property of convergence, is an asymptotic characteristic of the sequence of errors; it does not say *when* the announced rate of convergence occurs, i.e., what are the values of C or/and "large enough" values of t mentioned in the corresponding definitions. For concrete methods, the bounds on the indicated quantities typically can be extracted from the convergence proofs, but it does not help a lot – the bounds are usually very complicated, rough and depend on "invisible" quantitative characteristics of the problem like the magnitudes of high-order

derivatives, condition numbers of Hessians, etc. From these observations combined with the fact that our life is finite it follows that *one should not overestimate the importance of the rate-of-convergence ranking of the methods*. This traditional approach gives a kind of orientation, nothing more; unfortunately, there seems to be no purely theoretical way to get detailed ranking of numerical optimization methods. As a result, practical recommendations on which method to use are based on different theoretical and empirical considerations: theoretical rate of convergence, actual behaviour on test problems, numerical stability, simplicity and robustness, etc.

5.1.4 Global and Local solutions

The crucial intrinsic difficulty in Nonlinear Optimization is that *we cannot expect a numerical optimization method to approximate a globally optimal solution to the problem*.

The difficulty has its roots in *local* nature of the information on the problem which is available to the methods. Assume, e.g., that we are minimizing the function shown on the picture:



The function has two local minimizers, x' and x''. Observing small enough neighbourhood of every one of these minimizers, it is impossible to guess that in fact there exists another one. As a result, any "normal" method of nonlinear optimization as applied to the objective in question and being started from a small neighbourhood of the "wrong" (local, not global) minimizer x', will converge to x' – the local information on f available for the method does not allow to guess that x'' exists!

It would be wrong to say that the difficulty is absolutely unavoidable. We could run the method for different starting points, or even could look through the values of the objective along a sequence which is dense in $\mathbf{R}$ [1)] and define x_t as the best, in terms of the values of f, of the first t points of the sequence. The latter "method" can be easily extended to general constrained multidimensional problems; one can immediately prove its convergence to the global solution; the method is simple in implementation, etc. There is only one small drawback in the method: the tremendous number of function evaluations required to solve a problem within inaccuracy ϵ.

It can be easily seen that the outlined "method", as applied to a problem

$$\min_x \left\{ f(x) : x \in \mathbf{R}^n, g_1(x) = |x|^2 \leq 1 \right\}$$

with Lipschitz continuous, with constant 1, objective f:

$$|f(x) - f(y)| \leq |x - y|,$$

requires, *in the worst case*, at least ϵ^{-n} steps to find a point x_ϵ with the residual in terms of the objective – the quantity $f(x_\epsilon) - \min\limits_{|x|\leq 1} f$ – not exceeding ϵ.

[1)] i.e., visits arbitrary small neighbourhood of every point in $\mathbf{R}$, as it does, e.g., the sequence of all rational numbers (to arrange rationals in a single sequence, list them according to the sum of absolute values of the numerator and denominator in the corresponding fractions: first those with the above sum equal to 1 (the only rational 0=0/1), then those with the sum 2 (-1=-1/1,1=1/1), then those with the sum 3 (-2/1,-1/2,1/2,2/1), etc.)

> When $\epsilon = 0.01$ and $n = 20$ (very modest accuracy and dimension requirements), the number of steps becomes $> 10^{40}$, and this is the *lower* complexity bound!
>
> Moreover, for the family of problems in question the lower bound ϵ^{-n} on the number of function evaluations required to *guarantee* residual $\leq \epsilon$ is valid for *an arbitrary* optimization method which uses only local information on the objective.

Thus, we can approximate, within any given error $\epsilon > 0$, *the global* solution to any optimization problem; but to say that *the best* we can promise is to do this, for $\epsilon = 0.01$, $n = 20$, in 10^{40} steps – is worse than to say nothing.

As a consequence of the above considerations (same as of other, more advanced results of *Information-Based Complexity Theory of Optimization*), we come to the following important, although a desperate, conclusion:

> *It makes no sense to expect an optimization method to be able to approximate, with a reasonable inaccuracy in a reasonable time, global solution to all optimization problems of a given, even moderate, size.*

In fact all we may hope to do in reasonable time is to find tight approximations to certain (not necessarily corresponding to the optimal solution) Karush-Kuhn-Tucker point of an optimization problem (in the unconstrained case – to a critical point of the objective). In simple cases we may hope also to approximate a locally optimal solution, without any guarantees of its global optimality.

There is, anyhow, a "solvable case" when we can approximate with a reasonable complexity *globally optimal* solution to an optimization problem. This is the case when the problem is *convex* (i.e., the functions f and g_i, $i = 1, ..., m$, are convex, while h_j, if any, are linear). Properties of convex optimization problems and the numerical methods for these problems form the subject of Convex Programming. Convex Programming is, in its nature, simpler and, consequently, much more advanced than the general Nonlinear Optimization. In particular, in Convex Programming we are able to point out methods with quite reasonable *global* (not asymptotical!) rate of convergence which are capable to *guarantee*, at a reasonable computational cost, high-accuracy approximations of *globally* optimal solutions even for general-type convex programs.

I would be happy to restrict the remainder of our course with the nice world of Convex Programming, but we cannot afford it to ourselves: in actual applications we, unfortunately, too often meet with nonconvex problems, and we have no choice but to solve them – even at the cost of weakening the notion of "optimal solution" from the global to a local one (or even to the one of a Karush-Kuhn-Tucker point).

5.2 Line Search

The rest of the Lecture is devoted to *one-dimensional* unconstrained optimization, i.e., to numerical methods for solving problems

$$\min_x \{f(x) : x \in \mathbf{R}\}, \tag{5.2.1}$$

f being *at least* continuous function on the axis; these methods usually are called *line search*.

Our interest in line search comes, of course, not from the fact that in applications there are many one-dimensional problems, but rather from the fact that line search is a component of basically all traditional methods for multidimensional optimization. Typically, the scheme of a multidimensional method for unconstrained minimization is as follows: looking at local behaviour of the objective f at the current iterate x_t, the method chooses current "direction of movement" d_t (which, normally, is a descent direction of the objective: $d_t^T \nabla f(x_t) < 0$) and then performs a step in this direction:

$$x_t \mapsto x_{t+1} = x_t + \alpha_t d_t$$

in order to achieve certain progress in the objective value, i.e., to ensure that $f(x_{t+1}) < f(x_t)$. And in the majority of methods the step in the direction d_t is chosen by one-dimensional minimization of the function

$$\phi(\alpha) = f(x_t + \alpha d_t).$$

Thus, line search technique is crucial for basically all multidimensional optimization methods.

5.2.1 Zero-Order Line Search

We start with the *zero-order* line search, i.e., with methods for solving (5.2.1) which use the values of f only, not the derivatives.

The methods we are about to develop solve not the problem (5.2.1) as it is, but the problem

$$\min_x \{f(x) : a \le x \le b\} \tag{5.2.2}$$

of minimizing the objective over a given finite ($-\infty < a < b < \infty$) segment $[a, b]$ of the real axis. To ensure well-posedness of the problem, we make the following assumption:

f is *unimodal* on $[a, b]$, i.e., possesses a unique local minimum x^* on the segment.

This assumption, as it is easily seen, implies that f is strictly decreasing in $[a, b]$ to the left of x^*:

$$a \le x' < x'' \le x^* \Rightarrow f(x') > f(x'') \tag{5.2.3}$$

and is strictly increasing in $[a, b]$ to the right of x^*:

$$x^* \le x' < x'' \le b \Rightarrow f(x') < f(x''). \tag{5.2.4}$$

Indeed, if (5.2.3) were false, there would exist x' and x'' such that

$$a \le x' < x'' \le x^*, \;\; f(x') \le f(x'').$$

It follows that the set of minimizers of f on $[a, x'']$ contains a minimizer, x_*, which differs from x'' [2]. Since x_* is a minimizer of f on $[a, x'']$ and x_* differs from x'', x_* is a local minimizer of f on $[a, b]$, while it was assumed that the only local minimizer of f on $[a, b]$ is x^*; this gives the desired contradiction. Verification of (5.2.4) is similar.

Note that relations (5.2.3) - (5.2.4), in turn, imply that f is unimodal on $[a, b]$ and even on every smaller segment $[a', b'] \subset [a, b]$.

Given that f is unimodal on $[a, b]$, we immediately can point out a strategy for approximating x^*, namely, as follows. Let us choose somehow two points x^- and x^+ in (a, b),

$$a < x^- < x^+ < b,$$

and let us compute the values of f at these points. The basic observation is that

if [case A] $f(x^-) \le f(x^+)$, *then* x^* *is to the left of* x^+ [indeed, if x^* were to the right of x^+, then we would have $f(x^-) > f(x^+)$ by (5.2.3)], *and if* [case B] $f(x^-) \ge f(x^+)$, *then* x^* *is to the right of* x^- ["symmetric" reasoning].

Consequently, in the case of A we can replace the initial "uncertainty segment" $\Delta_0 = [a, b]$ with the new segment $\Delta_1 = [a, x^+]$, and in the case of B – with the segment $\Delta_1 = [x^-, b]$; in both cases the new "uncertainty segment" Δ_1 covers x^* and is strictly less than Δ_0. Since, at is was already mentioned, the objective, being unimodal on the initial segment $\Delta_0 = [a, b]$, is unimodal also on the smaller segment $\Delta_1 \subset \Delta_0$, we may iterate this procedure – choose two points in Δ_1, compute the values of the objective at these points, compare the results and replace Δ_1 with smaller uncertainty segment Δ_2, still containing the desired solution x^*, and so on.

Thus, we come to the following

Algorithm 5.2.1 [Conceptual zero-order minimization of unimodal function on $[a, b]$]
Initialization: *Set* $\Delta_0 = [a, b]$, $t = 1$
Step t: *Given previous uncertainty segment* $\Delta_{t-1} = [a_{t-1}, b_{t-1}]$,

[2] look: if x'' itself is not a minimizer of f on $[a, x'']$, then *any* minimizer of f on $[a, x'']$ can be chosen as x_*; if x'' is a minimizer of f on $[a, x'']$, then x' also is a minimizer, since $f(x') \le f(x'')$, and we can set $x_* = x'$

- *choose search points* x_t^-, x_t^+*:* $a_{t-1} < x_t^- < x_t^+ < b_{t-1}$*;*
- *compute* $f(x_t^-)$ *and* $f(x_t^+)$*;*
- *define the new uncertainty segment as follows: in the case of* $f(x_t^-) \le f(x_t^+)$ *set* $\Delta_t = [a_{t-1}, x_t^+]$*, otherwise set* $\Delta_t = [x_t^-, b_{t-1}]$*;*
- *replace* t *with* $t+1$ *and loop.*

It is immediately seen that we may ensure linear convergence of the lengths of subsequent uncertainty segments to 0, thus coming to a linearly converging algorithm for approximating x^*. For example, if x_t^-, x_t^+ are chosen to split Δ_{t-1} it into three equal parts, we ensure $|\Delta_{t+1}| = \frac{2}{3}|\Delta_t|$ ($|\Delta|$ stands for the length of a segment Δ), which results in linearly converging, with the convergence ratio $\sqrt{2/3}$, algorithm:

$$|x^* - x^k| \le \left(\frac{2}{3}\right)^{\lfloor k/2 \rfloor} |b-a|, \tag{5.2.5}$$

k being the # of function evaluations performed so far and x^k being an arbitrary point of the uncertainty segment $\Delta_{\lfloor k/2 \rfloor}$ formed after k function evaluations.

Estimate (5.2.5) is fine – we have non-asymptotical linear convergence rate with problem-independent convergence ratio. Could there be something better?

The answer is "yes". The way to improve the rate of convergence is to note that one of the two search points used to pass from Δ_t to Δ_{t+1} will for sure be inside Δ_{t+1}, and we could try to make it the search point used to pass from Δ_{t+1} to Δ_{t+2}; with this strategy, *the cost of updating* Δ_t *into* Δ_{t+1} *will be one function evaluation, not two of them* (except the very first updating $\Delta_0 \to \Delta_1$ which still will costs two function evaluations). There are two ways to implement this new smart strategy – the optimal one ("Fibonacci search") and the suboptimal ("Golden search").

5.2.1.1 Fibonacci search

The Fibonacci search can be used when we know in advance the number $N > 2$ of function evaluations we are going to perform.

Given N, consider the sequence of the first $N+1$ *Fibonacci integers* $F_0, F_1, F_2, ..., F_N$ defined by the recurrence

$$F_0 = F_1 = 1; F_k = F_{k-1} + F_{k-2}$$

(the first 10 integers in the sequence are $1, 1, 2, 3, 5, 8, 13, 21, 34, 55$). The method is as follows: given $\Delta_0 = [a, b]$, we set

$$d_0 = |b-a|,$$

choose two first search points x_1^- and x_1^+ at the distance

$$d_1 = \frac{F_{N-1}}{F_N} d_0$$

from the right and from the left endpoints of Δ_0, respectively (since $F_N/F_{N-1} = (F_{N-1} + F_{N-2})/F_{N-1} = 1 + F_{N-2}/F_{N-1} < 2$, we have $d_1 > d_0/2$, so that $x_1^- < x_1^+$). The length of the new uncertainty segment Δ_1 clearly will be d_1.

What we are about to do is to iterate the above step, with N replaced by $N-1$. Thus, now we should evaluate f at two points x_2^-, x_2^+ of the segment Δ_1 placed at the distance

$$d_2 = \frac{F_{N-2}}{F_{N-1}} d_1 \quad \left[= \frac{F_{N-2}}{F_{N-1}} \frac{F_{N-1}}{F_N} d_0 = \frac{F_{N-2}}{F_N} d_0\right] \tag{5.2.6}$$

from the right and the left endpoint of Δ_1. The crucial fact (which takes its origin in the arithmetic properties of the Fibonacci numbers) is that

one of these two required points where f should be computed is already processed – this is the one of the previous two points which belongs to Δ_1.

Indeed, assume, without loss of generality, that $\Delta_1 = [a, x_1^+]$ (the case $\Delta_1 = [x_1^-, b]$ is completely similar), so that the one of the first two search point belonging to Δ_1 is x_1^-. We have

$$x_1^- - a = (b - d_1) - a = (b - a) - d_1 = d_0 - d_1 = d_0\left(1 - \frac{F_{N-1}}{F_N}\right) =$$

[since $F_N = F_{N-1} + F_{N-2}$ and $d_2 = \frac{F_{N-2}}{F_N} d_0$]

$$= d_0 \frac{F_{N-2}}{F_N} = d_2.$$

Thus, only one of the two required points of Δ_1 is new for us, and another comes from the previous step; consequently, in order to update Δ_1 into Δ_2 we need one function evaluation, not two of them. After this new function evaluation, we are able to replace Δ_1 with Δ_2. To process Δ_2, we act exactly as above, but with N replaced by $N-2$; here we need to evaluate f at the two points of Δ_2 placed at the distance

$$d_3 = \frac{F_{N-3}}{F_{N-2}} d_2 \quad [= \frac{F_{N-3}}{F_N} d_0, \text{ see (5.2.6)}]$$

from the endpoints of the segment, and again one of these two points already is processed.

Iterating this procedure, we come to the segment Δ_{N-1} which covers x^*; the length of the segment is

$$d_{N-1} = \frac{F_1}{F_N} d_0 = \frac{b-a}{F_N},$$

and the total # of evaluations of f required to get this segment is N (we need 2 evaluations of f to pass from Δ_0 to Δ_1 and one evaluation per every of $N-2$ subsequent updatings $\Delta_t \mapsto \Delta_{t+1}$, $1 \le t \le N-2$).

Taking, as approximation of x^*, any point x^N of the segment Δ_{N-1}, we have

$$|x^N - x^*| \le |\Delta_N| = \frac{b-a}{F_N}. \tag{5.2.7}$$

To compare (5.2.7) with the accuracy estimate (5.2.5) of our initial – unsophisticated – method, note that

$$F_t = \frac{1}{\lambda + 2}\left[(\lambda+1)\lambda^t + (-1)^t \lambda^{-t}\right], \ \lambda = \frac{1+\sqrt{5}}{2} > 1.^{3)} \tag{5.2.8}$$

[3]Here is the computation: the Fibonacci numbers satisfy the homogeneous finite-difference equation

$$x_t - x_{t-1} - x_{t-2} = 0$$

and initial conditions $x_0 = x_1 = 1$. To solve a finite difference homogeneous equation, one should first look for its *fundamental solutions* – those of the type $x_t = \lambda^t$. Substituting $x_t = \lambda^t$ into the equation, we get a quadratic equation for λ:

$$\lambda^2 - \lambda - 1 = 0,$$

and we come to two fundamental solutions:

$$x_t^{(i)} = \lambda_i^t, \ i = 1, 2, \text{ with } \lambda_1 = \frac{1+\sqrt{5}}{2} > 1, \ \lambda_2 = -1/\lambda_1.$$

Any linear combination of these fundamental solutions again is a solution to the equation, and to get $\{F_t\}$, it remains to choose the coefficients of the combination to fit the initial conditions $F_0 = F_1 = 1$. As a result, we come to (5.2.8). A surprise is that the expression for *integer* quantities F_t involves irrational number!

Consequently, (5.2.7) results in

$$|x^N - x^*| \leq \frac{\lambda+2}{\lambda+1}\lambda^{-N}|b-a|(1+o(1)), \tag{5.2.9}$$

where $o(1)$ denotes a function of N which converges to 0 as $N \to \infty$).

We see that the convergence ratio for the Fibonacci search is

$$\lambda^{-1} = \frac{2}{1+\sqrt{5}} = 0.61803...$$

which is much better than the ratio $\sqrt{2/3} = 0.81649...$ given by (5.2.5).

It can be proved that the Fibonacci search is, in certain precise sense, the *optimal*, in terms of the accuracy guaranteed after N function evaluations, zero-order method for minimizing an unimodal function on a segment. In spite of this fine theoretical property, the method is not that convenient from the practical viewpoint: we should choose in advance the number of function evaluations to be performed (i.e., to tune the method to certain chosen in advance accuracy), which is sometimes inconvenient. The *Golden search* method we are about to present is free of this shortcoming and at the same time is, for not too small N, basically as efficient as the original Fibonacci search.

The idea of the Golden Search method is very simple: at k-th step of the N-step Fibonacci search, we choose two search points in the segment Δ_{k-1}, and each of these points divides the segment (from the closer endpoint to the more far one) in the ratio

$$[1 - F_{N-k}/F_{N-k+1}] : [F_{N-k}/F_{N-k+1}],$$

i.e., in the ratio $F_{N-k-1} : F_{N-k}$. According to (5.2.8), this ratio, for large $N-k$, is close to $1:\lambda$, $\lambda = (1+\sqrt{5})/2$. In the Golden search we use this ratio at *every* step, and that is it!

5.2.1.2 Golden search

Let $\lambda = (1+\sqrt{5})/2$ (this is the so called "golden ratio") . In the Golden search implementation of Algorithm 5.2.1 we choose at every step the search points x_t^- and x_t^+ to divide the previous segment of uncertainty $\Delta_{t-1} = [a_{t-1}, b_{t-1}]$ in the ratio $1:\lambda$:

$$x_t^- = \frac{\lambda}{1+\lambda}a_{t-1} + \frac{1}{1+\lambda}b_{t-1}; \quad x_t^+ = \frac{1}{1+\lambda}a_{t-1} + \frac{\lambda}{1+\lambda}b_{t-1}. \tag{5.2.10}$$

It is easily seen that for $t \geq 2$, one of the search points required to update Δ_{t-1} into Δ_t is already processed in course of updating Δ_{t-2} into Δ_{t-1}. To verify it, it suffices to consider the case when $\Delta_{t-2} = [\alpha, \beta]$ and $\Delta_{t-1} = [\alpha, x_{t-1}^+]$ (the "symmetric" case $\Delta_{t-1} = [x_{t-1}^-, \beta]$ is completely similar). Denoting $d = \beta - \alpha$, we have

$$x_{t-1}^- = \alpha + \frac{1}{1+\lambda}d, \;\; x_{t-1}^+ = \alpha + \frac{\lambda}{1+\lambda}d; \tag{5.2.11}$$

now, we are in situation $\Delta_{t-1} = [\alpha, x_{t-1}^+]$, so that the second of the two search points needed to update Δ_{t-1} into Δ_t is

$$x_t^+ = \alpha + \frac{\lambda}{1+\lambda}(x_{t-1}^+ - \alpha) = \alpha + \frac{\lambda^2}{(1+\lambda)^2}d$$

(see the second equality in (5.2.11)). The latter quantity, due to the first equality in (5.2.11) and the characteristic equation $\lambda^2 = 1+\lambda$ giving λ, is nothing but x_{t-1}^-:

$$\lambda^2 = 1+\lambda \Leftrightarrow \frac{1}{1+\lambda} = \frac{\lambda^2}{(1+\lambda)^2}.$$

Thus, in the Golden search each updating $\Delta_{t-1} \mapsto \Delta_t$, except the very first one, requires a single function evaluation, not two of them. The length of the uncertainty segment is reduced by every updating by factor

$$\frac{\lambda}{1+\lambda} = \frac{1}{\lambda},$$

i.e.,

$$|\Delta_t| = \lambda^{-t}(b-a).$$

After $N \geq 2$ function evaluations (i.e., after $t = N-1$ steps of the Golden search) we can approximate x^* by (any) point x^N of the resulting segment Δ_{N-1}, and inaccuracy bound will be

$$|x^N - x^*| \leq |\Delta_{N-1}| \leq \lambda^{1-N}(b-a). \tag{5.2.12}$$

Thus, we have the linear rate of convergence with convergence ratio $\lambda^{-1} = 0.61803...$, same as for the Fibonacci search, but now the method is "stationary" – we can perform as many steps of it as we wish.

5.2.2 Bisection

The theoretical advantage of the zero-order methods, like the Fibonacci search and the Golden search, is that these methods use the minimal information on the objective – its values only. Besides this, the methods have a very wide field of applications – the only requirement imposed on the objective is to be unimodal on a given segment which localizes the minimizer to be approximated. And even under these extremely mild restrictions these methods are linearly converging with objective-independent converging ratio; moreover, the corresponding *efficiency estimates* (5.2.9) and (5.2.12) are non-asymptotical: they do not contain "uncertain" constant factors and are valid for all values of N. At the same time, typically our objective is better behaved than a general unimodal function, e.g., is smooth enough. Making use of these additional properties of the objective, we may improve the behaviour of the line search methods.

Let us look what happens if we are solving problem (5.2.2) with *smooth* – continuously differentiable – objective. Same as above, assume that the objective is unimodal on $[a,b]$. In fact we make a little bit stronger assumption:

(A): *the minimizer x^* of f on $[a,b]$ is an interior point of the segment, and $f'(x)$ changes its sign at x^*:*

$$f'(x) < 0, x \in [a, x^*); \quad f'(x) > 0, x \in (x^*, b]$$

[unimodality + differentiability imply only that $f'(x) \leq 0$ on $[a,x^*)$ and $f'(x) \geq 0$ on $(x^*,b]$].

Besides these restrictions on the problem, assume, as it is normally the case, that we are able to compute not only the value, but also the derivative of the objective at a given point.

Under these assumptions we can solve (5.2.2) by definitely the simplest possible method – the *bisection*. Namely, let us compute f' at the midpoint x_1 of $\Delta_0 = [a,b]$. There are three possible cases:

- $f'(x_1) > 0$. This case, according to (A), is possible if and only if $x^* < x_1$, and we can replace the initial segment of uncertainty with $\Delta_1 = [a, x_1]$, thus reducing the length of the uncertainty segment by factor 2;
- $f'(x_1) < 0$. Similarly to the previous case, this inequality is possible if and only if $x^* > x_1$, and we can replace the initial segment of uncertainty with $[x_1, b]$, again reducing the length of the uncertainty segment by factor 2;
- $f'(x_1) = 0$. According to (A), it is possible if and only if $x_1 = x^*$, and we can terminate with exact minimizer at hand.

In the first two cases our objective clearly possesses property (A) with respect to the new segment of uncertainty, and we can iterate the construction. Thus, we come to

Algorithm 5.2.2 [Bisection]
Initialization: *set* $\Delta_0 = [a,b]$, $t = 1$
Step t: *Given previous uncertainty segment* $\Delta_{t-1} = [a_{t-1}, b_{t-1}]$,

- *define current search point x_t as the midpoint of Δ_{t-1}:*
$$x_t = \frac{a_{t-1} + b_{t-1}}{2};$$
- *compute $f'(x_t)$;*
- *in the case of $f'(x_t) = 0$ terminate and claim that x_t is the exact solution to (5.2.2). Otherwise set*
$$\Delta_t = \begin{cases} [a_{t-1}, x_t], & f'(x_t) > 0 \\ [x_t, b_{t-1}], & f'(x_t) < 0 \end{cases}$$
replace t with $t+1$ and loop.

From the above considerations we immediately conclude that

Proposition 5.2.1 [Linear convergence of Bisection]
Under assumption (A), for any $t \geq 1$, either the Bisection search terminates in course of the first t steps with exact solution x^*, or t-th uncertainty segment Δ_t is well-defined, covers x^* and is of the length $2^{-t}(b-a)$.

Thus, the Bisection method converges linearly with convergence ratio 0.5.

Remark 5.2.1 The convergence ratio of the Bisection algorithm is better than the one 0.61803... for Fibonacci/Golden search. There is no contradiction with the announced optimality of the Fibonacci search: the latter is optimal among the *zero-order* methods for minimizing unimodal functions, while Bisection is a first-order method.

Remark 5.2.2 The Bisection method can be viewed as the "limiting case" of the conceptual zero-order Algorithm 5.2.1: when, in the latter algorithm, we make both the search points x_t^- and x_t^+ close to the midpoint of the uncertainty segment Δ_{t-1}, the result of comparison between $f(x_t^-)$ and $f(x_t^+)$ which governs the choice of the new uncertainty segment in Algorithm 5.2.1 is given by the sign of f' at the midpoint of Δ_{t-1}.

Remark 5.2.3 Note that the assumption (A) can be weakened. Namely, let us assume that f' changes its sign at the segment $[a, b]$: $f'(a) < 0$, $f'(b) > 0$; and we assume nothing about the derivative in (a, b), except its continuity. In this case we still can successfully use the Bisection method to approximate a *critical point* of f in (a, b), i.e., a point where $f'(x) = 0$. Indeed, from the description of the method it is immediately seen that what we do is generating a sequence of nested segments $\Delta_0 \supset \Delta_1 \supset \Delta_2 \supset ...$, with the next segment being twice smaller than the previous one, with the property that f' changes it sign from $-$ to $+$ when passing from the left endpoint of every segment Δ_t to its right endpoint. This process can be terminated only in the case when the current iterate x_t is a critical point of f. If it does not happen, then the nested segments Δ_t have a unique common point x^*, and since in any neighbourhood of the point there are points both with positive and negative values of f', we have $f'(x^*) = 0$ (recall that f' is continuous). This is the critical point of f to which the algorithm converges linearly with convergence ratio 0.5.

The indicated remark explains the nature of the bisection algorithm. This is an algorithm for finding zero of the function f' rather than for minimizing f itself (under assumption (A), of course, this is the same - to minimize f on $[a, b]$ or to find the zero of f' on (a, b)). And the idea of the algorithm is absolutely trivial: given that the zero of f' is *bracketed* by the initial uncertainty segment $\Delta_0 = [a, b]$ (i.e., that f' at the endpoints of the segment is of different sign), we generate the sequence of enclosed segments, also bracketing zero of f', as follows: we split the previous segment $\Delta_t = [a_{t-1}, b_{t-1}]$ by its midpoint x_t into two subsegments $[a_{t-1}, x_t]$ and $[x_t, b_{t-1}]$. Since f' changes its sign when passing from a_{t-1} to b_{t-1}, it changes its sign either when passing from a_{t-1} to x_t, or when passing from x_t to b_{t-1} (provided that $f'(x_t) \neq 0$, so that we can speak about the sign of $f'(x_t)$; if $f'(x_t) = 0$, we are already done). We detect on which of the two subsegments f' in fact changes sign and take it as the new uncertainty segment Δ_t; by construction, it also brackets a zero of f'.

5.2.3 Curve fitting

The line search methods considered so far possess, under unimodality assumption, nice *objective-independent global linear convergence*. May we hope for something better? Of course, yes: it would be fine to get something superlinearly converging. If the objective is "well-behaved" – smooth enough – we have good chances to accelerate convergence, at least at the final stage, by *curve fitting*, i.e., by approximating the objective by a simple function with analytically computable minimum. The natural way is to approximate f by a polynomial, choosing the coefficients of the polynomial in order to fit it to the observed values (and derivatives, if we are able to compute these derivatives) of f at several "most perspective" iterates. An iteration of a pure curve fitting algorithm is as follows:

- at the beginning of the iteration, we have certain set of "working points" where we already have computed the values and, possibly, certain derivatives of the objective. Given these data, we compute the *current approximating polynomial* p which should have the same values and derivatives at the working points as those of the objective
- after approximating polynomial p is computed, we find analytically its minimizer and take it as the new search point
- we compute the value (and, possibly, the derivatives) of the objective at the new search point and update the set of working points, adding to it the last search point (along with the information on the objective at this point) and excluding from this set the "worst" of the previous working points, and then loop.

The idea underlying the outlined approach is very simple: if we somehow can enforce the procedure to converge, the working points will eventually be at certain small distance d from the minimizer of f. If f is smooth enough, the error in approximating f by p in the d-neighbourhood of working points will be of order of d^{q+1}, q being the degree of p, and the error in approximating f' by p' will be of order of d^q. Consequently, we may hope that the distance from the minimizer of p (i.e., the zero of p') to the minimizer of f (the zero of f') will be of order of d^q, which gives us good chances for superlinear convergence.

Of course, what is said is nothing but a very rough idea. Let us look at several standard implementations.

5.2.3.1 Newton's method

Assume that we are solving problem (5.2.1) with twice continuously differentiable objective f, and that, given x, we can compute $f(x)$, $f'(x)$, $f''(x)$. Under these assumptions we can apply to the problem the *Newton* method as follows:

Algorithm 5.2.3 [One-dimensional Newton method]
Initialization: *choose somehow starting point* x_0
Step t: *given the previous iterate* x_{t-1},

- *compute* $f(x_{t-1}), f'(x_{t-1}), f''(x_{t-1})$ *and approximate* f *around* x_{t-1} *by its second-order Taylor expansion*
$$p(x) = f(x_{t-1}) + f'(x_{t-1})(x - x_{t-1}) + \frac{1}{2}f''(x_{t-1})(x - x_{t-1})^2;$$
- *choose as* x_t *the minimizer of the quadratic function* $p(\cdot)$*:*
$$x_t = x_{t-1} - \frac{f'(x_{t-1})}{f''(x_{t-1})},$$
replace t *with* $t+1$ *and loop.*

The Newton method, *started close to a nondegenerate local minimizer* x^* *of* f (i.e., close to a point x^* satisfying the sufficient second order optimality condition: $f'(x^*) = 0$, $f''(x^*) > 0$), converges to x^* quadratically:

Proposition 5.2.2 [Local quadratic convergence of the Newton method]
Let $x^* \in \mathbf{R}$ be a nondegenerate local minimizer of smooth function f, i.e., a point such that f is three times continuously differentiable in a neighbourhood of x^* with $f'(x^*) = 0$, $f''(x^*) > 0$. Then the Newton iterates converge to x^* quadratically, provided that the starting point x_0 is close enough to x^*.

Proof. Let $g(x) = f'(x)$, so that $g(x^*) = 0$, $g'(x^*) > 0$ and

$$x_t = x_{t-1} - \frac{g(x_{t-1})}{g'(x_{t-1})}.$$

Since $g = f'$ is twice continuously differentiable in a neighbourhood of x^* and $g'(x^*) > 0$, there exist positive constants k_1, k_2, r such that

$$|x' - x^*|, |x'' - x^*| \le r \Rightarrow |g'(x') - g'(x'')| \le k_1 |x' - x''|, \;\; g'(x') \ge k_2. \tag{5.2.13}$$

Now let

$$\rho = \min\{r; \frac{k_2}{k_1}\}. \tag{5.2.14}$$

Assume that for certain t the iterate x_{t-1} belongs to the ρ-neighbourhood

$$U_\rho = [x^* - \rho, x^* + \rho]$$

of the point x^*. Then $g'(x_{t-1}) \ge k_2 > 0$ (due to (5.2.13); note that $\rho \le r$), so that the Newton iterate x_t of x_{t-1} is well-defined. We have

$$x_t - x^* = x_{t-1} - x^* - \frac{g(x_{t-1})}{g'(x_{t-1})} =$$

[since $g(x^*) = 0$]

$$x_{t-1} - x^* - \frac{g(x_{t-1}) - g(x^*)}{g'(x_{t-1})} = \frac{g(x^*) - g(x_{t-1}) - g'(x_{t-1})(x^* - x_{t-1})}{g'(x_{t-1})}.$$

The numerator in the resulting fraction is the remainder in the first order Taylor expansion of g at x_{t-1}; due to (5.2.13) and since $|x_{t-1} - x^*| \le \rho \le r$, it does not exceed in absolute value the quantity $\frac{1}{2} k_1 |x^* - x_{t-1}|^2$. The denominator, by the same (5.2.13), is at least k_2. Thus,

$$x_{t-1} \in U_\rho \Rightarrow |x_t - x^*| \le \frac{k_1}{2k_2} |x_{t-1} - x^*|^2. \tag{5.2.15}$$

Due to the origin of ρ, (5.2.15) implies that

$$|x_t - x^*| \le |x_{t-1} - x^*|/2;$$

we see that the trajectory of the Newton method, once reaching U_ρ, never leaves this neighbourhood and converges to x^* linearly with convergence ratio 0.5. It for sure is the case when $x_0 \in U_\rho$, and let us specify the "close enough" in the statement of the proposition just as inclusion $x_0 \in U_\rho$. With this specification, we get that the trajectory converges to x^* linearly, and from (5.2.15) it follows that in fact the order of convergence is (at least) 2. ∎

Remark 5.2.4 Both the assumptions that $f''(x^*) > 0$ and that x_0 is close enough are essential[4]. For example, as applied to the smooth convex function

$$f(x) = x^4$$

[4] in fact, the assumption $f''(x^*) > 0$ can be replaced with $f''(x^*) < 0$, since the trajectory of the method remains unchanged when f is replaced with $-f$ (in other words, the Newton method does not distinguish between the local minima and local maxima of the objective). We are speaking about the case of $f''(x^*) > 0$, not the one of $f''(x^*) < 0$, simply because the former case is the only important for minimization.

(with degenerate minimizer $x^* = 0$), the method becomes

$$x_t = x_{t-1} - \frac{1}{3}x_{t-1} = \frac{2}{3}x_{t-1};$$

in this example the method converges, but linearly rather than quadratically.

As applied to strictly convex smooth function

$$f(x) = \sqrt{1+x^2}$$

with unique (and nondegenerate) local (and global as well) minimizer $x^* = 0$, the method becomes, as it is immediately seen,

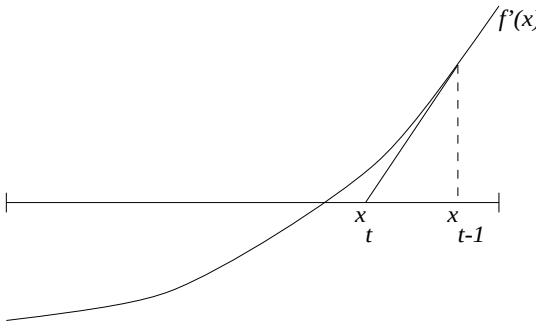
$$x_t = -x_{t-1}^3;$$

this procedure converges (very fast: with order 3) to 0 provided that the starting point is in $(-1, 1)$, and diverges to infinity – also very fast – if $|x_0| > 1$.

In fact the Newton method is a *linearization method for finding zero of f'*: given the previous iterate x_{t-1}, we linearize $g = f'$ at this point and take as x_t the solution to the linearization

$$g(x_{t-1}) + g'(x_{t-1})(x - x_{t-1}) = 0$$

of the actual equation $g(x) = 0$.



Newton method as zero-finding routine

5.2.3.2 Regula Falsi (False Position) method

This method, same as the Newton one, is based on approximating f by a quadratic polynomial, but here this polynomial is constructed via two working points with first-order information rather than via a single working point with second-order information. The method, in its most straightforward form, is as follows. Given two latest iterates x_{t-1}, x_{t-2}, along with the values f and f' at these iterates, we approximate $f''(x_{t-1})$ by the finite difference

$$\frac{f'(x_{t-1}) - f'(x_{t-2})}{x_{t-1} - x_{t-2}}$$

and use this approximation to approximate f by a quadratic function

$$p(x) = f(x_{t-1}) + f'(x_{t-1})(x - x_{t-1}) + \frac{1}{2}\frac{f'(x_{t-1}) - f'(x_{t-2})}{x_{t-1} - x_{t-2}}(x - x_{t-1})^2.$$

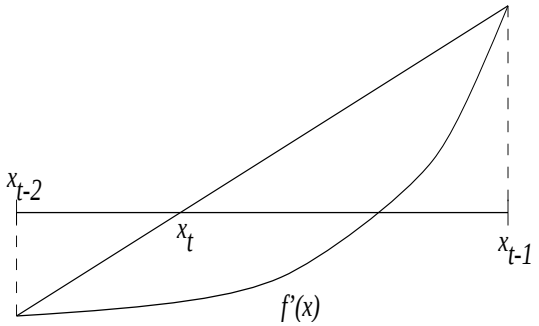
The new iterate is the minimizer of this quadratic function:

$$x_t = x_{t-1} - f'(x_{t-1})\frac{x_{t-1} - x_{t-2}}{f'(x_{t-1}) - f'(x_{t-2})}. \tag{5.2.16}$$

Note that although the polynomial p is chosen in asymmetric with respect to x_{t-1} and x_{t-2} way (it is tangent to f at x_{t-1}, but even not necessarily coincides with f at x_{t-2}), the minimizer x_t of this polynomial is symmetric with respect to the pair of working points; as it is immediately seen, the right hand side of (5.2.16) can be equivalently rewritten as

$$x_t = x_{t-2} - f'(x_{t-2})\frac{x_{t-1} - x_{t-2}}{f'(x_{t-1}) - f'(x_{t-2})}.$$

The geometry of the method is very simple: same as the Newton method, this is the method which actually approximates the zero of $g(x) = f'(x)$ (look: the values of f are not involved into the recurrence (5.2.16)). In the Newton method we, given the value and the derivative of g at x_{t-1}, approximate the graph of g by its tangent at x_{t-1} line $g(x_{t-1}) + g'(x_{t-1})(x - x_{t-1})$ and choose x_t as the point where this tangent line crosses the x-axis. In the Regula Falsi method we, given the values of g at two points x_{t-1} and x_{t-2}, approximate the graph of g by the secant line passing through $(x_{t-1}, g(x_{t-1}))$ and $(x_{t-2}, g(x_{t-2}))$ and choose as x_t the point where this secant line crosses the x-axis.



Regula Falsi method as zero-finding routine

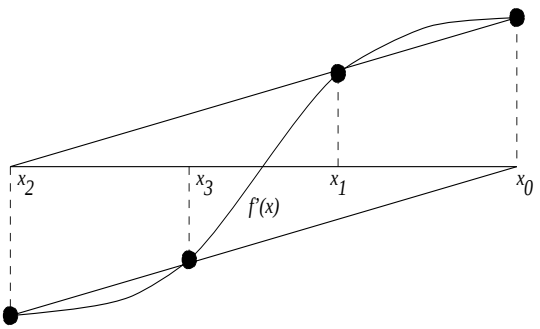
The local rate of convergence of the method is given by the following

Proposition 5.2.3 [Local superlinear convergence of Regula Falsi method]

Let x^* be a nondegenerate minimizer of a smooth function f, namely, a point such that f is three times continuously differentiable in a neighbourhood of x^* with $f'(x^*) = 0$, $f''(x^*) > 0$. Then, for starting pair x_0, x_1 of two distinct points close enough to x^* the method converges to x^* superlinearly with order of convergence

$$\lambda = \frac{1 + \sqrt{5}}{2}.$$

Note that, same as for the Newton method, the assumptions of the proposition, especially the one of closeness of x_0 and x_1 to x^*, are essential: badly started, the method may be non-converging:



Cycle in Regula Falsi: the trajectory is $x_0, x_1, x_2, x_3, x_0, x_1, ...$

5.2.3.3 Cubic fit

Approximating polynomials used in curve fitting methods are of small order – not greater than 3, which is clear: to run the method, we should be able to compute the minimizer of a polynomial analytically. The *Cubic fit* is based on this "highest-order" approximation. At the step t of the method, given the latest two iterates $x' = x_{t-1}$ and $x'' = x_{t-2}$ along with the values of f and f' at the points, one defines the cubic polynomial $p(\cdot)$ such that

$$p(x') = f(x'),\ p'(x') = f'(x'),\ p(x'') = f(x''),\ p'(x'') = f'(x'')$$

(these are four linear equations on four coefficients of the polynomial; if $x' \neq x''$, which, as we shall see, always can be assumed, the equations uniquely define p). As the next iterate, one chooses the local minimizer of p. If exists, the minimizer is given by the relations

$$x_t = x' - (x' - x'')\left[\frac{u_1 + u_2 - f'(x')}{f'(x'') - f'(x') + 2u_2}\right],$$

$$u_1 = f'(x') + f'(x'') - 3\frac{f(x') - f(x'')}{x' - x''},\ \ u_2 = \sqrt{u_1^2 - f'(x')f'(x'')}. \tag{5.2.17}$$

The step is for sure well-defined if $f'(x')$ and $f'(x'')$ are of opposite signs ("V-shape"; compare with Bisection). One can prove that if x^* is a nondegenerate local minimizer of a smooth enough function f, then the method, *started close enough to* x^*, converges to x^* quadratically.

5.2.3.4 Safeguarded curve fitting

There are many other curve fitting schemes: the one based on rational approximation of the objective, the one where the objective is approximated by a quadratic polynomial according to its values at three-point working set, etc. All curve fitting methods have common advantage: superlinear local convergence for well-behaved functions. And all these methods have common disadvantage as well: if the method is started far from optimum, the method can diverge (see, e.g., Remark 5.2.4). To overcome this severe shortcoming, it is recommended to combine reliable linearly converging (in the case of unimodal objectives) methods like Golden search or Bisection with curve fitting. The idea is as follows. At step t of the method we have previous uncertainty segment $\Delta_{t-1} = [a_{t-1}, b_{t-1}]$ ($f'(x_{t-1}) < 0$, $f'(b_{t-1}) > 0$), as well as certain working set W_t comprised of several previous iterates where we know the values of f and, possibly, the derivatives of f. Same as in the curve fitting methods, we use the working set to form polynomial (or rational) approximation p to f and compute analytically the minimizer of p (or something close to this minimizer). In the curve fitting methods the resulting point, let it be called u, is used as the next iterate. In contrast to this, in the safeguarded curve fitting it is used as the next iterate only if u is "reasonably placed" with respect to Δ_{t-1}; if it is not the case, the next iterate is chosen according to the Bisection/Golden search scheme, depending on whether we use the derivatives of f.

The notion of a "reasonably placed" u contains several tests. First of all, u should be well-defined and belong to Δ_{t-1}, but this is not all. It is recommended, e.g., not to use u if it is too far from the best (with the smallest value of f) point of the working set, say, is at the distance larger than $\frac{1}{2}|\Delta_{t-1}|$ (since in this case the Bisection step is likely to reduce inaccuracy better than the curve fitting one). After the step – the curve fitting or the "safeguarding" (Bisection/Golden search) one – is performed, one updates the working set and the uncertainty segment.

Of course, what is said is not a description of a concrete method, but a very rough idea which admits many implementations (detailed specification of which curve fitting method to use, what is a "reasonably placed" u, rules for updating the working set, etc.) With good implementation, the safeguarded curve fitting combines the advantages of the Bisection/Golden

search (global linear convergence with objective-independent rate in the unimodal case) and those of curve fitting (superlinear local convergence for well-behaved objectives).

5.2.4 Inexact Line Search

As it was already mentioned, the main application of line search methods is inside algorithms for multi-dimensional optimization. In these algorithms we always allow only small number of steps of the line search subroutine at each iteration of the master algorithm, otherwise the overall complexity of the master method will be too large. Moreover, in many multidimensional algorithms we are not that interested in high-accuracy solutions of one-dimensional subproblems; what is crucial for the master method, is reasonable progress in solving these subproblems. Whenever it is the case, we can terminate line search relatively far from the actual solution of the subproblem in question, using certain simple tests for "reasonable progress". Let us describe two most popular tests of this type.

5.2.4.1 Armijo's rule

Consider the situation which is typical for application of line search technique inside multi-dimensional master method. At an iteration of the latter method we have *current iterate* $x \in \mathbf{R}^n$ and *search direction* $d \in \mathbf{R}^n$ which is a *descent* direction for our multivariate objective $f(\cdot) : \mathbf{R}^n \to \mathbf{R}$:

$$d^T \nabla f(x) < 0. \tag{5.2.18}$$

The goal is to reduce "essentially" the value of the objective by a step

$$x \mapsto x + \gamma^* d$$

from x in the direction d.

Assume that f is continuously differentiable. Then the function

$$\phi(\gamma) = f(x + \gamma d)$$

of one variable also is once continuously differentiable; moreover, due to (5.2.18), we have

$$\phi'(0) < 0,$$

so that for small positive γ one has

$$\phi(\gamma) - \phi(0) \approx \gamma \phi'(0) < 0.$$

Our desire is to choose a "'reasonably large" stepsize $\gamma^* > 0$ which results in the progress $\phi(\gamma^*) - \phi(0)$ in the objective "of order of $\gamma^* \phi'(0)$". The Armijo test for this requirement is as follows:

Armijo's Test:
we fix once for ever constants $\epsilon \in (0, 1)$ (popular choice is $\epsilon = 0.2$) and $\eta > 1$ (say, $\eta = 2$ or $\eta = 10$) and say that candidate value $\gamma > 0$ is appropriate, if the following two relations are satisfied:

$$\phi(\gamma) \le \phi(0) + \epsilon \gamma \phi'(0) \tag{5.2.19}$$

[this part of the test says that the progress in the value of ϕ given by the stepsize γ is "of order of $\gamma \phi'(0)$"]

$$\phi(\eta\gamma) \ge \phi(0) + \epsilon \eta \gamma \phi'(0) \tag{5.2.20}$$

[this part of the test says that γ is "maximal in order" stepsize which satisfies (5.2.19) – if we multiply γ by η, the increased value fails to satisfy (5.2.19), at least to satisfy it as a strict inequality]

Under assumption (5.2.18) and the additional (very natural) assumption that f (and, consequently, ϕ) is below bounded, the Armijo test is *consistent*: there do exist values of $\gamma > 0$ which pass the test. To see it, it suffices to notice that

A. (5.2.19) is satisfied for all small enough positive γ.
Indeed, since ϕ is differentiable, we have

$$0 > \phi'(0) = \lim_{\gamma \to +0} \frac{\phi(\gamma) - \phi(0)}{\gamma},$$

whence

$$\epsilon\phi'(0) \geq \frac{\phi(\gamma) - \phi(0)}{\gamma}$$

for all small enough positive γ (since $\epsilon\phi'(0) > \phi'(0)$ due to $\phi'(0) < 0, \epsilon \in (0,1)$); the resulting inequality is equivalent to (5.2.19);

B. (5.2.19) is not valid for all large enough values of γ.
Indeed, the right hand side of (5.2.19) tends to $-\infty$ as $\gamma \to \infty$, due to $\phi'(0) < 0$, and the left hand side is assumed to be below bounded.

Now let us choose an arbitrary positive $\gamma = \gamma_0$ and test whether it satisfies (5.2.19). If it is the case, let us replace this value subsequently by $\gamma_1 = \eta\gamma_0$, $\gamma_2 = \eta\gamma_1$, etc., each time verifying whether the new value of γ satisfies (5.2.19). According to **B**, this cannot last forever: for certain $s \geq 1$ γ_s for sure fails to satisfy (5.2.19). When it happens for the first time, the quantity γ_{s-1} turns out to satisfy (5.2.19), while the quantity $\gamma_s = \eta\gamma_{s-1}$ fails to satisfy (5.2.19), which means that $\gamma = \gamma_{s-1}$ passes the Armijo test.

If the initial γ_0 does not satisfy (5.2.19), we replace this value subsequently by $\gamma_1 = \eta^{-1}\gamma_0$, $\gamma_2 = \eta^{-1}\gamma_1$, etc., each time verifying whether the new value of γ still does not satisfy (5.2.19). According to **A**, this cannot last forever: for certain $s \geq 1$, γ_s for sure satisfies (5.2.19). When it happens for the first time, γ_s turns out to satisfy (5.2.19), while $\gamma_{s-1} = \eta\gamma_s$ fails to satisfy (5.2.19), and $\gamma = \gamma_s$ passes the Armijo test.

Note that the presented proof in fact gives an explicit (and fast) algorithm for finding a stepsize passing the Armijo test, and this algorithm can be used (and often is used) in Armijo-aimed line search instead of more accurate (and normally more time-consuming) line search methods from the previous sections.

5.2.4.2 Goldstein test

Another popular test for "sufficient progress" in line search is as follows:

Goldstein test:
we fix one for ever constant $\epsilon \in (0, 1/2)$ and say that candidate value $\gamma > 0$ is appropriate, if

$$\phi(0) + (1-\epsilon)\gamma\phi'(0) \leq \phi(\gamma) \leq \phi(0) + \epsilon\gamma\phi'(0). \tag{5.2.21}$$

Here again relation (5.2.16) and below boundedness of f imply consistency of the test.

Lecture 6

Gradient Descent and Newton Methods

Starting from this lecture, we shall speak about methods for solving unconstrained multidimensional problems

$$f(x) \to \min \mid \; x \in \mathbf{R}^n. \tag{6.0.1}$$

From now on, let us make the following assumptions:

- (A) the objective f in (6.0.1) is continuously differentiable;
- (B) the problem in question is solvable: the set

$$X^* = \operatorname*{Argmin}_{\mathbf{R}^n} f$$

  is nonempty.

6.1 Gradient Descent

This section is devoted to the oldest and most widely known method for (6.0.1) - the *Gradient Descent*.

6.1.1 The idea

The idea of the method is very simple. Assume that we are at certain point x, and that we have computed $f(x)$ and $\nabla f(x)$. Assume that x is not a critical point of f: $\nabla f(x) \neq 0$ (this is the same as to say that x is not a Karush-Kuhn-Tucker point of the problem). Then $g = -\nabla f(x)$ is a *descent* direction of f at x:

$$\frac{d}{d\gamma}|_{\gamma=0} f(x - \gamma \nabla f(x)) = -|\nabla f(x)|^2 < 0;$$

moreover, this is the *best* among the descent directions h (normalized to have the same length as that one of g) of f at x: for any h, $|h| = |g|$, one has

$$\frac{d}{d\gamma}|_{\gamma=0} f(x + \gamma h) = h^T \nabla f(x) \geq -|h||\nabla f(x)| = -|\nabla f(x)|^2$$

(we have used the Cauchy inequality), the inequality being equality if and only if $h = g$.

The indicated observation demonstrates that in order to improve x – to form a new point with smaller value of the objective – it makes sense to perform a step

$$x \mapsto x + \gamma g \equiv x - \gamma \nabla f(x)$$

from x in the intergradient direction; with properly chosen stepsize $\gamma > 0$, such a step will for sure decrease f. And in the Gradient Descent method, we simply iterate the above step. Thus, the generic scheme of the method is as follows

Algorithm 6.1.1 [Generic Gradient Descent]
Initialization: *choose somehow starting point* x_0 *and set* $t = 1$
Step t: *at the beginning of step* t *we have previous iterate* x_{t-1}. *At the step we*

- *compute* $f(x_{t-1})$ *and* $\nabla f(x_{t-1})$;
- *choose somehow a positive stepsize* γ_t, *set*

$$x_t = x_{t-1} - \gamma_t \nabla f(x_{t-1}), \tag{6.1.1}$$

replace t *with* $t+1$ *and loop.*

Thus, the generic Gradient Descent method is the recurrence (6.1.1) with certain rule for choosing stepsizes $\gamma_t > 0$; normally, the stepsizes are given by a kind of line search applied to the univariate functions

$$\phi_t(\gamma) = f(x_{t-1} - \gamma \nabla f(x_{t-1})).$$

6.1.2 Standard implementations

Different versions of line search result in different versions of the Gradient Descent method. Among these versions, one should mention

- ArD [Gradient Descent with Armijo-terminated line search]: the stepsize $\gamma_t > 0$ at iteration t where $\nabla f(x_{t-1}) \neq 0$ is chosen according to the Armijo test (Section 5.2.4.1):

$$f(x_{t-1} - \gamma_t \nabla f(x_{t-1})) \leq f(x_{t-1}) - \epsilon \gamma_t |\nabla f(x_{t-1})|^2;$$

$$f(x_{t-1} - \eta \gamma_t \nabla f(x_{t-1})) \geq f(x_{t-1}) - \epsilon \eta \gamma_t |\nabla f(x_{t-1})|^2, \tag{6.1.2}$$

  $\epsilon \in (0,1)$ and $\eta > 1$ being the parameters of the method. And if x_{t-1} is a critical point of f, i.e., $\nabla f(x_{t-1}) = 0$, the choice of $\gamma_t > 0$ is absolutely unimportant: independently of the value of γ_t, (6.1.1) will result in $x_t = x_{t-1}$.

- StD [Steepest Descent]: γ_t minimizes f along the ray $\{x_{t-1} - \gamma \nabla f(x_{t-1}) \mid \gamma \geq 0\}$:

$$\gamma_t \in \operatorname*{Argmin}_{\gamma \geq 0} f(x_{t-1} - \gamma \nabla f(x_{t-1})). \tag{6.1.3}$$

Of course, the Steepest Descent is a kind of idealization: in nontrivial cases we are unable to minimize the objective along the search ray *exactly*. Moreover, to make this idealization valid, we should assume that the corresponding steps are well-defined, i.e., that

$$\operatorname*{Argmin}_{\gamma \geq 0} f(x - \gamma \nabla f(x)) \neq \emptyset$$

for every x; in what follows, this is assumed "by default" whenever we are speaking about the Steepest Descent.

In contrast to the Steepest Descent, the Gradient Descent with Armijo-terminated line search is quite "constructive" – we know from Section 5.2.4.1 how to find a stepsize passing the Armijo test.

6.1.3 Convergence of the Gradient Descent

6.1.3.1 General Convergence Theorem

We start with the following theorem which establishes, under very mild restrictions, global convergence of the Gradient Descent to the set of *critical points* of f – to the set

$$X^{**} = \{x \in \mathbf{R}^n \mid \nabla f(x) = 0\}.$$

Theorem 6.1.1 [Global convergence of Gradient Descent] *For both* StD *and* ArD *, the following statements are true:*

(i) *If the trajectory $\{x_t\}$ of the method is bounded, then the trajectory possesses limiting points, and all these limiting points are critical points of f;*

(ii) *If the level set*

$$S = \{x \in \mathbf{R}^n \mid f(x) \leq f(x_0)\}$$

of the objective is bounded, then the trajectory of the method is bounded (and, consequently, all its limiting points, by (i)*, belong to X^{**}).*

Proof. (ii) is an immediate consequence of (i), since both ArD and StD clearly are *descent* methods:

$$x_t \neq x_{t-1} \Rightarrow f(x_t) < f(x_{t-1}). \tag{6.1.4}$$

Therefore the trajectory, for each of the methods, is contained in the level set S; since under assumption of (ii) this set is bounded, the trajectory also is bounded, as claimed in (ii).

It remains to prove (i). Thus, let the trajectory $\{x_t\}$ be bounded, and let x^* be a limiting point of the trajectory; we should prove that $\nabla f(x^*) = 0$. Assume, on contrary, that it is not the case, and let us lead this assumption to a contradiction. The idea of what follows is very simple: since $\nabla f(x^*) \neq 0$, a step of the method taken from x^* reduces f by certain positive amount δ; this is absolutely clear from the construction of the step. What is very likely (it should, of course, be proved, and we shall do it in a while) is that *there exists a small neighbourhood U of x^* such that a step of the method taken from arbitrary point $x \in U$ also improves the objective at least by fixed positive quantity δ'*. It is absolutely unimportant for us what is this δ'; all we need is to know that this quantity is positive and is independent of the particular choice of $x \in U$. Assume that we already have proved that the required U and δ' exist. With this assumption, we get the desired contradiction immediately: since x^* is a limiting point of the trajectory, the trajectory visits U infinitely many times. Each time it visits U, the corresponding step decreases f at least by $\delta' > 0$, and no step of the method increases the objective. Thus, in course of running the method we infinitely many times decrease the objective by δ' and never increase it, so that the objective must diverge to $-\infty$ along our trajectory; the latter is impossible, since the objective was assumed to be below bounded.

Now it is time to prove our key argument – the one on existence of the above U and δ'. Let me stress why there is something to be proved, in spite of the already known to us descentness of the method – the fact that the objective is improved by every step taken from a non-critical point of f (and all points close enough to non-critical x^* also are noncritical, since ∇f is continuous). The difficulty is that the progress in f in course of a step depends on from which point the step is taken; in principle it might happen that a step from every point from a neighbourhood of x^* improves the objective, but there is no *independent of the point positive* lower bound δ' for the improvements. And in the above reasoning we indeed require "point-independent" progress – otherwise it might happen that subsequent visits of U by the trajectory result in smaller and smaller improvements in f, and the sum of these improvements is finite; this possibility would kill the above reasoning completely.

In fact, of course, the required U, δ' exist. It suffices to prove this statement for ArD only – it is absolutely clear that the progress in the objective in course of a step of StD is at least the one for a step of ArD , both steps being taken from the same point. The proof for the case of ArD looks as follows:

Since f is continuously differentiable and $\nabla f(x^*) \neq 0$, there exist positive r, P and p such that

$$|x - x^*| < r \Rightarrow p \leq |\nabla f(x)| \leq P;$$

by the same reasons, there exists $r' \in (0, r)$ such that in the r'-neighbourhood V of x^* one has

$$|\nabla f(x') - \nabla f(x'')| \leq \zeta \equiv (1-\epsilon)P^{-1}p^2.$$

Let U be $r'/2$-neighbourhood of x^*. We claim that

(*) *whenever $x \in U$, the stepsize s_x given by the Armijo line search as applied to the function*

$$\phi_x(s) = f(x - s\nabla f(x)) \quad [\phi_x'(0) = -|\nabla f(x)|^2]$$

is at least

$$s^* = \frac{1}{2} r' \eta^{-1} P^{-1}.$$

Note that (*) is all we need. Indeed, the progress in the objective in the Armijo line search as applied to a function ϕ and resulting in a stepsize s is at least $\epsilon s|\phi'(0)|$. Applying this observation to a step of ArD taken from a point $x \in U$ and using (*), we come to the conclusion that the progress in the objective at the step is at least $\epsilon s^* |\nabla f(x)|^2 \geq \epsilon s^* p^2$, and the latter quantity (which is positive and is independent of $x \in U$) can be taken as the desired δ'.

It remains to prove (*), which is immediate: assuming that $x \in U$, $s_x < s^*$, and taking into account the construction of the Armijo test, we would get

$$\phi_x(\eta s_x) - \phi_x(0) > \epsilon \eta s_x \phi'(0). \tag{6.1.5}$$

Now, since $s_x < s^*$, the segment $[x, x - \eta s_x \nabla f(x)]$ is of the length at most $\eta s^* P \leq r'/2$, and since one endpoint of the segment belongs to U, the segment itself belongs to V. Consequently, the derivative of f along the segment varies at most by ζ, so that the derivative of ϕ varies on the segment $[0, \eta s_x]$ at most by

$$|\nabla f(x)|\zeta \leq P\zeta = (1-\epsilon)p^2.$$

On the other hand, from the Lagrange Mean Value Theorem it follows that

$$\phi(\eta s_x) - \phi(0) = \eta s_x \phi'(\xi) \leq \eta s_x \phi'(0) + \eta s_x (1-\epsilon)p^2;$$

here ξ is some point on the segment $[0, \eta s_x]$. Combining this inequality with (6.1.5), we come to

$$\eta s_x (1-\epsilon)p^2 > -(1-\epsilon)\eta s_x \phi'(0) \equiv (1-\epsilon)\eta s_x |\nabla f(x)|^2 \geq (1-\epsilon)\eta s_x p^2,$$

which is a contradiction. ∎

Please pay attention to the above proof: it structure is typical for convergence proofs in traditional Optimization and looks as follows. We know in advance that the iterative process in question possesses certain *Lyapunov function* L – one which decreases along the trajectory of the process and is below bounded (in the above proof this function is f itself); we also either assume that the trajectory is bounded, or assume boundedness of the level set of the Lyapunov function, the set being associated with the value of the function at the initial point of the trajectory (then, of course, the trajectory for sure is bounded – since the Lyapunov function never decreases along the trajectory, the latter is unable to leave the aforementioned level set). Now assume that the three entities – (1) the Lyapunov function, (2) our iterative process, and (3) the set X^* we agree to treat as the solution set of our problem – are linked by the following relation:

(**) *if a point on the trajectory does not belong to X^*, then the step of the process from this point strictly decreases the Lyapunov function*

Normally (**) is evident from the construction of the process and of the Lyapunov function; e.g., in the above proof where L is the objective, the process is ArD or StD and X^* is the set of critical points of the objective, you should not work hard in order to prove that the step from a non-critical point somehow decreases the objective. Now, given all this, we are interested to prove that the trajectory of the process converges to X^*; what is the main point of the proof? Of course, an analogy of (*), i.e., a "locally uniform" version of (**) – we should prove that a point not belonging to X^* possesses a neighbourhood such that whenever the trajectory visits this neighbourhood, the progress in the Lyapunov function at the corresponding step is bounded away from zero. After we have proved this crucial fact, we can immediately apply the scheme of the above proof to demonstrate that the trajectory indeed converges to X^*.

We have had a good reason to invest that many effort in explaining the "driving forces" of the above convergence proof: from now on, we shall skip similar proofs, since we believe that the reader understands the general principle, and the technicalities are of minor interest here. We hope that now it becomes clear why in the Armijo test we require the stepsize to be the largest one (up to factor η) resulting in "significant" progress in the objective. If we would skip this "maximality" requirement, we would allow arbitrarily small stepsizes even from the points which are far from the solution set; as a result, (*) would not be the case anymore, and we would become unable to ensure convergence of the process (and this property indeed may be lost).

6.1.3.2 Limiting points of Gradient Descent

We have proved that the standard versions of the Gradient Descent, under the assumption that the trajectory is bounded, converge to the set X^{**} of critical points of the objective. This set for sure contains the set X^* of global minimizers of f, same as the set of local minimizers of the objective, but this is not all: X^{**} contains also all local maximizers of f and the saddle points of the function, if any exists. An important question is whether a limiting point of the trajectory of the Gradient Descent can be something we are not interested in – a critical point which is *not* a local minimizer of the objective. What can be said is the following: a *nondegenerate* local maximizer x^* of f (i.e., a critical point of f such that $f''(x^*)$ is negative definite) *cannot* be a limiting point of the trajectories of ArD and StD , excluding the case when the x^* happens to be a point of the trajectory; this may happen in ArD (although it is "practically impossible"), and it never happens in StD , except the trivial (and also "practically impossible") case when the trajectory is started at x^*. And, speaking informally, it is "very unlikely" that a limiting point of the trajectory is a saddle point of the objective. Thus, "in practice" limiting points of the trajectory of Gradient Descent are local minimizers of the objective.

6.1.4 Rates of convergence

6.1.4.1 Rate of global convergence: general $C^{1,1}$ case

As we already know, under the assumption of item (ii) of Theorem 6.1.1 (i.e., when the level set $S = \{x \mid f(x) \leq f(x_0)\}$ is bounded), the versions of the Gradient Descent mentioned in the Theorem converge to the set X^{**} of critical points of f. What can be said about the *non-asymptotical* rate of convergence? The answer depends on how we measure the inaccuracy. If we use to this purpose something like the distance

$$\mathrm{dist}(x, X^{**}) = \min_{y \in X^{**}} |y - x|$$

from an approximate solution x to X^{**}, no nontrivial efficiency estimates can be established: the convergence of the quantities $\mathrm{dist}(x_t, X^{**})$ to 0 can be arbitrarily slow, even when f is convex. There is, however, another accuracy measure,

$$\epsilon_f(x) = |\nabla f(x)|^2,$$

more suitable for our purposes. Note that the set X^{**} towards which the trajectory converges is exactly the set where $\epsilon_f(\cdot) = 0$, so that $\epsilon_f(x)$ indeed can be viewed as something which measures the "residual in the inclusion $x \in X^{**}$". And it turns out that we can point out the rate at which this residual converges to 0:

Proposition 6.1.1 [Non-asymptotical rate of convergence of Gradient Descent]
Assume that the objective f is a $C^{1,1}$ function, i.e., it is continuously differentiable with Lipschitz continuous gradient:

$$|\nabla f(x) - \nabla f(y)| \leq L_f |x - y|, \quad \forall x, y \in \mathbf{R}^n. \tag{6.1.6}$$

Then for any integer $N > 0$:

(i) *For the started at x_0 trajectory $\{x_t\}$ of* StD *one has*

$$\epsilon_f[t] \equiv \min_{0 \leq t < N} |\nabla f(x_t)|^2 \leq \frac{2L_f}{N}[f(x_0) - \min f]. \tag{6.1.7}$$

(ii) *For the started at x_0 trajectory $\{x_t\}$ of* ArD *one has*

$$\epsilon_f[t] \equiv \min_{0 \leq t < N} |\nabla f(x_t)|^2 \leq \frac{\eta L_f}{2\epsilon(1-\epsilon)N}[f(x_0) - \min f], \tag{6.1.8}$$

$\epsilon \in (0, 1)$, $\eta > 1$ being the parameters of the underlying Armijo test.

Proof.

1^0. Let us start with the following simple

Lemma 6.1.1 *Under assumption of the Theorem one has*

$$f(y) \leq f(x) + (y - x)^T \nabla f(x) + \frac{L_f}{2}|y - x|^2, \quad \forall x, y \in \mathbf{R}^n. \tag{6.1.9}$$

Proof of Lemma. Let $\phi(\gamma) = f(x + \gamma(y - x))$. Note that ϕ is continuously differentiable (since f is) and

$$|\phi'(\alpha) - \phi'(\beta)| = |(y - x)^T(\nabla f(x + \alpha(y - x)) - \nabla f(x + \beta(y - x))| \leq$$

[Cauchy's inequality]

$$\leq |y - x||\nabla f(x + \alpha(y - x)) - \nabla f(x + \beta(y - x))| \leq$$

[(6.1.6)]

$$\leq |y - x|^2 L_f |\alpha - \beta|.$$

Thus,

$$|\phi'(\alpha) - \phi'(\beta)| \leq L_f |y - x|^2 |\alpha - \beta|, \quad \forall \alpha, \beta \in \mathbf{R}. \tag{6.1.10}$$

We have

$$f(y) - f(x) - (y - x)^T \nabla f(x) = \phi(1) - \phi(0) - \phi'(0) = \int_0^1 \phi'(\alpha) d\alpha - \phi'(0) =$$

$$= \int_0^1 [\phi'(\alpha) - \phi'(0)] d\alpha \leq$$

[see (6.1.10)]

$$\leq \int_0^1 |y - x|^2 L_f \alpha d\alpha = \frac{L_f}{2}|y - x|^2,$$

as required in (6.1.9). ∎

2^0. Now we are ready to prove (i). By construction of the Steepest Descent,

$$f(x_t) = \min_{\gamma \geq 0} f(x_{t-1} - \gamma \nabla f(x_{t-1})) \leq$$

[by Lemma 6.1.1]

$$\leq \min_{\gamma\geq 0}\left[f(x_{t-1}) + [-\gamma\nabla f(x_{t-1})]^T\nabla f(x_{t-1}) + \frac{L_f}{2}|\gamma\nabla f(x_{t-1})|^2\right] =$$

$$= f(x_{t-1}) + |\nabla f(x_{t-1})|^2 \min_{\gamma\geq 0}\left[-\gamma + \frac{L_f}{2}\gamma^2\right] = f(x_{t-1}) - \frac{1}{2L_f}|\nabla f(x_{t-1})|^2.$$

Thus, we come to the following important inequality:

$$f(x_{t-1}) - f(x_t) \geq \frac{1}{2L_f}|\nabla f(x_{t-1})|^2 \tag{6.1.11}$$

– *the progress in the objective at a step of Steepest Descent is at least of order of the squared norm of the gradient at the previous iterate.*

To conclude the proof, it suffices to note that, due to the monotonicity of the method, the total progress in the objective in course of certain segment of steps cannot be more than the initial residual $f(x_0) - \min f$ in the objective value; consequently, in a long segment, there must be a step with small progress, i.e., with small norm of the gradient. To make this reasoning quantitative, let us take sum of inequalities (6.1.11) over $t = 1, ..., N$, coming to

$$\frac{1}{2L_f}\sum_{t=0}^{N-1}|\nabla f(x_t)|^2 \leq f(x_0) - f(x_N) \leq f(x_0) - \min f.$$

The left hand side here is $\geq \frac{N}{2L_f}\min_{0\leq t<N}|\nabla f(x_t)|^2$, and (6.1.7) follows. ∎

3^0. The proof of (ii) is a little bit more involved, but follows the same basic idea: the progress at a step of ArD can be small only if the gradient at the previous iterate is small, and the progress at certain step from a long segment of the steps must be small, since the total progress cannot be larger than the initial residual. Thus, in a long segment of steps we must pass through a point with small norm of the gradient.

The quantitative reasoning is as follows. First of all, progress in the objective at a step t of ArD is not too small, provided that both γ_t and $|\nabla f(x_{t-1})|^2$ are not too small:

$$f(x_{t-1}) - f(x_t) \geq \epsilon\gamma_t|\nabla f(x_{t-1})|^2; \tag{6.1.12}$$

this is immediate consequence of the first inequality in (6.1.2). Second, γ_t is not too small. Indeed, by Lemma 6.1.1 applied with $x = x_{t-1}, y = x_{t-1} - \eta\gamma_t\nabla f(x_{t-1})$ we have

$$f(x_{t-1} - \eta\gamma_t\nabla f(x_{t-1})) \leq f(x_{t-1}) - \eta\gamma_t|\nabla f(x_{t-1})|^2 + \frac{L_f}{2}\eta^2\gamma_t^2|\nabla f(x_{t-1})|^2,$$

while by the second inequality in (6.1.2)

$$f(x_{t-1} - \eta\gamma_t\nabla f(x_{t-1})) \geq f(x_{t-1}) - \epsilon\eta\gamma_t|\nabla f(x_{t-1})|^2.$$

Combining these inequalities, we get

$$(1-\epsilon)\eta\gamma_t|\nabla f(x_{t-1})|^2 \leq \frac{L_f}{2}\eta^2\gamma_t^2|\nabla f(x_{t-1})|^2.$$

Since $\gamma_t > 0$, in the case of $\nabla f(x_{t-1}) \neq 0$ we obtain

$$\gamma_t \geq \frac{2(1-\epsilon)}{\eta L_f}; \tag{6.1.13}$$

in the case of $\nabla f(x_{t-1}) = 0$ γ_t, as we remember, can be chosen in arbitrary way without influencing the trajectory (the latter in any case will satisfy $x_{t-1} = x_t = x_{t+1} = ...$), and we may assume that γ_t *always* satisfies (6.1.13).

Combining (6.1.12) and (6.1.13), we come to the following inequality (compare with (6.1.11):

$$f(x_{t-1}) - f(x_t) \geq \frac{2\epsilon(1-\epsilon)}{\eta L_f} |\nabla f(x_{t-1})|^2. \tag{6.1.14}$$

Now the proof can be completed exactly as in the case of the Steepest Descent. ■

Remark 6.1.1 The efficiency estimate of Proposition 6.1.1 gives sublinearly converging to 0 *non-asymptotical* upper bound on the inaccuracies $\epsilon_f(\cdot)$ of the iterates. Note, anyhow, that this is a bound on the inaccuracy of *the best* (with the smallest norm of the gradient) of the iterates generated in course of the first N steps of the method, not on the inaccuracy of the *last* iterate x_N (the quantities $|\nabla f(x_t)|^2$ may oscillate, in contrast to the values $f(x_t)$ of the objective).

6.1.4.2 Rate of global convergence: convex $C^{1,1}$ case

Theorem 6.1.1 says that under mild assumptions the trajectories of ArD and StD converge to the set X^{**} of critical points of f. If we assume, in addition, that f is *convex*, so that the set of critical points of f is the same as the set of global minimizers of the function, we may claim that the trajectories converge to the optimal set of the problem. Moreover, in the case of convex $C^{1,1}$ objective (see Proposition 6.1.1) we can get non-asymptotical efficiency estimates in terms of the residuals $f(x_t) - \min f$, and under additional nondegeneracy assumption (see below) – also in terms of the distances $|x_t - x^*|$ from the iterates to the optimal solution.

To simplify our considerations and to make them more "practical", in what follows we restrict ourselves with the Armijo-based version ArD of the Gradient Descent.

Convex $C^{1,1}$ case:

Proposition 6.1.2 [Rate of global convergence of ArD in convex $C^{1,1}$ case]
Let the parameter ϵ in the ArD method be ≥ 0.5, and let f be a convex $C^{1,1}$ function with a nonempty set X^* of global minimizers. Then

(i) *The trajectory $\{x_t\}$ of ArD converges to certain point $x^* \in X^*$;*

(ii) *For every $N \geq 1$ one has*

$$f(x_N) - \min f \leq \frac{\eta L_f \operatorname{dist}^2(x_0, x^*)}{4(1-\epsilon)N}, \tag{6.1.15}$$

where L_f is the Lipschitz constant of $\nabla f(\cdot)$ and

$$\operatorname{dist}(x, X^*) = \min_{y \in X^*} |y - x|. \tag{6.1.16}$$

Proof.

1^0. Let x^* be a point from X^*, and let us look how the squared distances

$$d_t^2 = |x_t - x^*|^2$$

vary with t. We have

$$d_t^2 = |x_t - x^*|^2 \equiv |[x_{t-1} - \gamma_t \nabla f(x_{t-1})] - x^*|^2 = |[x_{t-1} - x^*] - \gamma_t \nabla f(x_{t-1})|^2 =$$
$$= |x_{t-1} - x^*|^2 - 2\gamma_t (x_{t-1} - x^*)^T \nabla f(x_{t-1}) + \gamma_t^2 |\nabla f(x_{t-1})|^2. \tag{6.1.17}$$

Since f is convex, from the Gradient Inequality

$$f(y) \geq f(x) + (y - x)^T \nabla f(x) \quad \forall x, y \in \mathbf{R}^n$$

it follows that

$$(x_{t-1} - x^*)^T \nabla f(x_{t-1}) \geq f(x_{t-1}) - f(x^*) = f(x_{t-1}) - \min f.$$

This inequality combined with (6.1.17) results in

$$d_t^2 \leq d_{t-1}^2 - \gamma_t \left[2\epsilon_{t-1} - \gamma_t |\nabla f(x_{t-1})|^2\right], \;\; \epsilon_s \equiv f(x_s) - \min f \geq 0. \tag{6.1.18}$$

According to (6.1.12), we have

$$\gamma_t |\nabla f(x_{t-1})|^2 \leq \frac{1}{\epsilon}[f(x_{t-1}) - f(x_t)] = \frac{1}{\epsilon}[\epsilon_{t-1} - \epsilon_t].$$

Combining this inequality with (6.1.18), we get

$$d_t^2 \leq d_{t-1}^2 - \gamma_t \left[(2 - \epsilon^{-1})\epsilon_{t-1} + \epsilon^{-1}\epsilon_t\right]. \tag{6.1.19}$$

Since, by assumption, $1/2 \leq \epsilon$, and clearly $\epsilon_s \geq 0$, the quantity in the parentheses in the right hand side is nonnegative. We know also from (6.1.13) that

$$\gamma_t \geq \bar{\gamma} = \frac{2(1-\epsilon)}{\eta L_f},$$

so that (6.1.19) results in

$$d_t^2 \leq d_{t-1}^2 - \bar{\gamma} \left[(2 - \epsilon^{-1})\epsilon_{t-1} + \epsilon^{-1}\epsilon_t\right]. \tag{6.1.20}$$

We conclude, consequently, that

(*) *The distances from the points x_t to (any) point $x^* \in X^*$ do not increase with t. In particular, the trajectory is bounded.*

From (*) it immediately follows that $\{x_t\}$ converges to certain point $\bar{x}^* \in X^*$, as claimed in (i). Indeed, by Theorem 6.1.1 the trajectory, being bounded, has all its limiting points in the set X^{**} of critical points of f, or, which is the same (f is convex!), in the set X^* of global minimizers of f. Let $\bar{x}^*$ be one of these limiting points, and let us prove that in fact $\{x_t\}$ converges to $\bar{x}^*$. To this end note that the sequence $|x_t - \bar{x}^*|$, which, as we know from (*), is non-increasing, has 0 as its limiting point; consequently, the sequence converges to 0, so that $x_t \to \bar{x}^*$ as $t \to \infty$, as claimed.

It remains to prove (6.1.15). Taking sum of inequalities (6.1.20) over $t = 1, ..., N$, we get

$$N\bar{\gamma} \left[(2 - \epsilon^{-1})\epsilon_{t-1} + \epsilon^{-1}\epsilon_t\right] \leq d_0^2 - d_N^2 \leq d_0^2 \equiv |x_0 - x^*|^2.$$

Since $\epsilon_0 \geq \epsilon_1 \geq \epsilon_2 \geq ...$ (our method is descent – it never decreases the values of the objective!), the left hand side in the resulting inequality can only become smaller if we replace all ϵ_t with ϵ_N; thus, we get

$$2N\bar{\gamma}\epsilon_N \leq |x_0 - x^*|^2, \tag{6.1.21}$$

whence, substituting the expression for $\bar{\gamma}$,

$$\epsilon_N \leq \frac{\eta L_f |x_0 - x^*|^2}{4(1-\epsilon)N};$$

since the resulting inequality is valid for all $x^* \in X^*$, (6.1.15) follows. ∎

Strongly convex $\mathbf{C}^{1,1}$ case. Proposition 6.1.2 deals with the case of *smooth convex* f, but there were no assumptions on the non-degeneracy of the minimizer – the minimizer might be non-unique, and the graph of f could be very "flat" around X^*. Under additional assumption of *strong convexity* of f we may get better convergence results.

The notion of strong convexity is given by the following

Definition 6.1.1 [Strongly convex function] *A function $f : \mathbf{R}^n \to \mathbf{R}$ is called strongly convex with the parameters of strong convexity (l_f, L_f), $0 < l_f \le L_f \le \infty$, if f is continuously differentiable and satisfies the inequalities*

$$f(x) + (y-x)^T \nabla f(x) + \frac{l_f}{2}|y-x|^2 \le f(y) \le f(x) + (y-x)^T \nabla f(x) + \frac{L_f}{2}|y-x|^2, \ \ \forall x, y \in \mathbf{R}^n. \quad (6.1.22)$$

Strongly convex functions traditionally play the role of "excellent" objectives, and this is the family on which the theoretical convergence analysis of optimization methods is normally performed. For our further purposes it is worthy to know how to detect strong convexity and what are the basic properties of strongly convex functions; this is the issue we are coming to.

The most convenient sufficient condition for strong convexity is given by the following

Proposition 6.1.3 [Criterion of strong convexity for twice continuously differentiable functions] *Let $f : \mathbf{R}^n \to \mathbf{R}$ be twice continuously differentiable function, and let (l_f, L_f), $0 < l_f \le L_f < \infty$, be two given reals. f is strongly convex with parameters l_f, L_f if and only if the spectrum of the Hessian of f at every point $x \in \mathbf{R}^n$ belongs to the segment $[l_f, L_f]$:*

$$l_f \le \lambda_{\min}(\nabla^2 f(x)) \le \lambda_{\max}(\nabla^2 f(x)) \le L_f \ \ \forall x \in \mathbf{R}^n, \quad (6.1.23)$$

where $\lambda_{\min}(A), \lambda_{\max}(A)$ denote the minimal, respectively, the maximal eigenvalue of a symmetric matrix A and $\nabla^2 f(x)$ denotes the Hessian (the matrix of the second order derivatives) of f at x.

Example 6.1.1 *The convex quadratic form*

$$f(x) = \frac{1}{2} x^T A x - b^T x + c,$$

A being positive definite symmetric matrix, is strongly convex with the parameters $l_f = \lambda_{\min}(A)$, $L_f = \lambda_{\max}(A)$.

The most important for us properties of strongly convex functions are summarized in the following statement:

Proposition 6.1.4 *Let f be strongly convex with parameters (l_f, L_f). Then*
(i) *The level sets $\{x \mid f(x) \le a\}$ of f are compact for every real a;*
(ii) *f attains its global minimum on $\mathbf{R}^n$, the minimizer being unique;*
(iii) *$\nabla f(x)$ is Lipschitz continuous with Lipschitz constant L_f.*

Now we come back to Gradient Descent. The following important proposition says that ArD as applied to a strongly convex f possesses global linear convergence:

Proposition 6.1.5 [Linear convergence of ArD as applied to strongly convex f] *Let a strongly convex, with parameters (l_f, L_f), function f be minimized by ArD started at certain point x_0, and let the parameter ϵ of the Armijo test underlying the method be $\ge 1/2$. Then, for every integer $N \ge 1$, one has*

$$|x_N - x^*| \le \theta^N |x_0 - x^*|, \quad \theta = \sqrt{\frac{Q_f - (2-\epsilon^{-1})(1-\epsilon)\eta^{-1}}{Q_f + (\epsilon^{-1}-1)\eta^{-1}}}, \quad (6.1.24)$$

where x^* is the (unique, due to Proposition 6.1.4.(ii)) minimizer of f and

$$Q_f = \frac{L_f}{l_f} \tag{6.1.25}$$

is the condition number of f.

Besides this,

$$f(x_N) - \min f \le \theta^{2N} Q_f [f(x_0) - \min f]. \tag{6.1.26}$$

Thus, the method globally linearly converges with convergence ratio θ (note that $\theta \in (0,1)$ due to $\epsilon \in [1/2, 1)$).

Proof.

1^0. According to Proposition 6.1.4, f is $C^{1,1}$ convex function which attains its minimum, and the gradient of f is Lipschitz continuous with constant L_f. Consequently, all conclusions of the proof of Proposition 6.1.2 are valid, in particular, relation (6.1.19):

$$d_t^2 \equiv |x_t - x^*|^2 \le d_{t-1}^2 - \bar\gamma \left[(2-\epsilon^{-1})\epsilon_{t-1} + \epsilon^{-1}\epsilon_t\right], \ \bar\gamma = \frac{2(1-\epsilon)}{\eta L_f}, \ \epsilon_s = f(x_s) - \min f. \tag{6.1.27}$$

Applying (6.1.22) to the pair $(x = x^*, y = x_s)$ and taking into account that $\nabla f(x^*) = 0$, we get

$$\epsilon_s \ge \frac{l_f}{2}|x_s - x^*|^2 = \frac{l_f}{2} d_s^2;$$

therefore (6.1.27) implies

$$d_t^2 \le d_{t-1}^2 - \frac{\bar\gamma l_f}{2}\left[(2-\epsilon^{-1})d_{t-1}^2 + \epsilon^{-1} d_t^2\right],$$

or, substituting the expression for $\bar\gamma$ and rearranging the expression,

$$d_t^2 \le \theta^2 d_{t-1}^2, \tag{6.1.28}$$

with θ given by (6.1.24), and (6.1.24) follows.

It remains to prove (6.1.26). To this end it suffices to note that, due to the first inequality in (6.1.22) applied with $x = x^*, y = x_0$, one has

$$|x_0 - x^*|^2 \le \frac{2}{l_f}[f(x_0) - f(x^*)] = \frac{2}{l_f}[f(x_0) - \min f], \tag{6.1.29}$$

while the second inequality in (6.1.22) applied with $x = x^*, y = x_N$ says that

$$f(x_N) - \min f \equiv f(x_N) - f(x^*) \le \frac{L_f}{2}|x_N - x^*|^2;$$

consequently,

$$f(x_N) - \min f \le \frac{L_f}{2}|x_N - x^*|^2 \le$$

[see (6.1.24)]

$$\le \frac{L_f}{2}\theta^{2N}|x_0 - x^*|^2 \le$$

[see (6.1.29)]

$$\le \frac{L_f}{l_f}\theta^{2N}[f(x_0) - \min f],$$

as required in (6.1.26). ∎

Global rate of convergence in convex $C^{1,1}$ case: summary. The results given by Propositions 6.1.2 and 6.1.5 can be summarized as follows. Assume that we are solving the problem

$$f(x) \to \min$$

with convex $C^{1,1}$ objective (i.e., $\nabla f(x)$ is a Lipschitz continuous vector field), and assume that f possesses a nonempty set X^* of global minimizers. And assume that we are minimizing f by ArD with properly chosen parameter ϵ, namely, $1/2 \le \epsilon < 1$. Then

- A. In the general case, where no strong convexity of f is imposed, the trajectory $\{x_t\}$ of the method converges to certain point $\bar{x}^* \in X^*$, and the residuals in terms of the objective – the quantities $\epsilon_N = f(x_N) - \min f$ – go to zero at least as $O(1/N)$, namely, they satisfy the estimate

$$\epsilon_N \le \frac{\eta L_f \operatorname{dist}^2(x_0, X^*)}{4(1-\epsilon)} \frac{1}{N}. \tag{6.1.30}$$

  Note that

  - no quantitative assertions on the rate of convergence of the quantities $|x_N - \bar{x}^*|$ can be given; all we know is that these quantities converge to 0, but the convergence can be as slow as you wish. Namely, given an arbitrary decreasing sequence $\{d_t\}$ converging to 0, one can point out a $C^{1,1}$ convex function f on 2D plane such that $L_f = 1$, $\operatorname{dist}(x_0, X^*) = d_0$ and $\operatorname{dist}(x_t, X^*) \ge d_t$ for every t;
  - estimate (6.1.30) establishes correct order of convergence to 0 of the residuals in terms of the objective: for properly chosen $C^{1,1}$ convex function f on the 2D plane one has

$$\epsilon_N \ge \frac{\alpha}{N}, \ N = 1, 2, ...$$

    with certain positive α.

- B. If f is strongly convex with parameters (l_f, L_f), then the method converges linearly:

$$|x_N - x^*| \le \theta^N |x_0 - x^*|, \ \ f(x_N) - \min f \le Q_f \theta^{2N} [f(x_0) - \min f],$$

$$\theta = \sqrt{\frac{Q_f - (2 - \epsilon^{-1})(1-\epsilon)\eta^{-1}}{Q_f + (\epsilon^{-1} - 1)\eta^{-1}}}, \tag{6.1.31}$$

  $Q_f = L_f / l_f$ being the condition number of f.

Note that the convergence ratio θ (or θ^2, depending on which accuracy measure – the distance from the iterate to the optimal set or the residual in terms of the objective – we use) tends to 1 as the condition number of the problem goes to infinity (as people say, as the problem becomes *ill-conditioned*). When Q_f is large, we have

$$\theta \approx 1 - p Q_f^{-1}, \ \ p = (1-\epsilon)\eta^{-1}, \tag{6.1.32}$$

so that to decrease the upper bound (6.1.31) on $|x_\cdot - x^*|$ by an absolute constant factor, say, by factor 10, it requires $O(Q_f)$ steps of the method. In other words, what we can extract from (6.1.31) is that

(**) *the number of steps of the method resulting in a given in advance progress in accuracy (the one required to decrease the initial distance from the optimal set by a given factor, say, 10^6), is proportional to the condition number Q_f of the objective.*

Of course, this conclusion is derived from an *upper* bound on inaccuracy; it might happen that our upper bounds "underestimate" actual performance of the method. It turns out, anyhow, that our bounds are tight, and the conclusion is valid:

> *the number of steps of the Gradient Descent required to reduce initial inaccuracy* (measured either as the distance from the optimal set or as the residual in terms of the objective) *by a given factor is typically proportional to the condition number of f.*

To justify the claim, let us look what happens in the case of *quadratic* objective.

6.1.4.3 Rate of convergence in the quadratic case

Let us look what happens if Gradient Descent is applied to a strongly convex quadratic objective

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c.$$

A being symmetric positive definite matrix. As we know from Example 6.1.1, f is strongly convex with the parameters $l_f = \lambda_{\min}(A)$, $L_f = \lambda_{\max}(A)$ (the minimal and the maximal eigenvalues of A, respectively).

It is convenient to speak about the Steepest Descent rather than about the Armijo-based Gradient Descent (in the latter case our considerations would suffer from uncertainty in the choice of the stepsizes).

We have the following relations:

- The gradient of the function f is given by the relation

$$g(x) \equiv \nabla f(x) = Ax - b; \tag{6.1.33}$$

  in particular, the unique minimizer of f is given by the equation (the Fermat rule)

$$Ax^* = b. \tag{6.1.34}$$

  Note also that, as it is seen from one-line computation,

$$f(x) = E(x) + f(x^*), \;\; E(x) = \frac{1}{2}(x - x^*)^T A(x - x^*); \tag{6.1.35}$$

  note that $E(\cdot)$ is nothing but inaccuracy in terms of the objective.

- The trajectory of the Steepest Descent is given by the recurrence

$$x_{t+1} = x_t - \gamma_{t+1} g_t, \;\; g_t \equiv g(x_t) \equiv \nabla f(x_t) = Ax_t - b = A(x_t - x^*), \tag{6.1.36}$$

  where γ_{t+1} is minimizer of the strongly convex quadratic function $\phi(\gamma) = f(x_t - \gamma g_t)$ of real variable γ. Solving equation $\phi'(\gamma) = 0$ which identifies γ_{t+1}, we get

$$\gamma_{t+1} = \frac{g_t^T g_t}{g_t^T A g_t}; \tag{6.1.37}$$

  thus, (6.1.36) becomes

$$x_{t+1} = x_t - \frac{g_t^T g_t}{g_t^T A g_t} g_t. \tag{6.1.38}$$

- Explicit computation results in [1)]

$$E(x_{t+1}) = \left\{1 - \frac{(g_t^T g_t)^2}{[g_t^T A g_t][g_t^T A^{-1} g_t]}\right\} E(x_t). \tag{6.1.39}$$

Now we can obtain the convergence rate of the method from the following

Lemma 6.1.2 [Kantorovich] *Let A be a positive definite symmetric matrix with the condition number (the ratio between the largest and the smallest eigenvalue) Q. Then for any nonzero vector x one has*

$$\frac{(x^T x)^2}{[x^T Ax][x^T A^{-1} x]} \geq \frac{4Q}{(1+Q)^2}.$$

Proof. It is known from Linear Algebra that a symmetric $n \times n$ matrix A is orthogonally equivalent to a diagonal matrix S (i.e., $A = USU^T$ with orthogonal U), the eigenvalues $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ of A being the diagonal entries of S. Denoting $y = U^T x$, we see that the left hand side in the inequality in question is

$$\frac{(\sum_i y_i^2)^2}{(\sum_i \lambda_i y_i^2)(\sum_i \lambda_i^{-1} y_i^2)}. \tag{6.1.40}$$

This quantity remains unchanged if all y_i's are multiplied by common nonzero factor; thus, without loss of generality we may assume that $\sum_i y_i^2 = 1$. Further, the quantity in question remains unchanged if all λ_i's are multiplied by common positive factor; thus, we may assume that $\lambda_1 = 1$, so that $\lambda_n = Q$ is the condition number of the matrix A. Setting $a_i = y_i^2$, we come to the necessity to prove that

if $u = \sum_i a_i \lambda_i$, $v = \sum_i a_i \lambda_i^{-1}$, where $0 \leq a_i$, $\sum_i a_i = 1$, and $1 \leq \lambda_i \leq Q$, then $uv \leq (1+Q)^2/(4Q)$.

This is easy: due to its origin, the point (u, v) on the 2D plane is convex combination, the coefficients being a_i, of the points $P_i = (\lambda_i, \lambda_i^{-1})$ belonging to the arc Γ on the graph of the function $\eta = 1/\xi$, the arc corresponding to the segment $[1, Q]$ of values of ξ (ξ, η are

[1] Here is the computation: since $\phi(\gamma)$ is a convex quadratic form and γ_{t+1} is its minimizer, we have

$$\phi(0) = \phi(\gamma_{t+1}) + \frac{1}{2}\gamma_{t+1}^2 \phi'';$$

due to the origin of ϕ, we have $\phi'' = g_t^T A g_t$, so that

$$E(x_t) - E(x_{t+1}) \equiv f(x_t) - f(x_{t+1}) \equiv \phi(0) - \phi(\gamma_{t+1}) = \frac{1}{2}\gamma_{t+1}^2 [g_t^T A g_t],$$

or, due to (6.1.37),

$$E(x_t) - E(x_{t+1}) = \frac{(g_t^T g_t)^2}{2 g_t^T A g_t}.$$

At the same time by (6.1.35), (6.1.36) one has

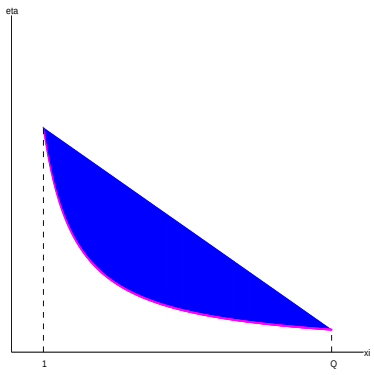
$$E(x_t) = \frac{1}{2}(x_t - x^*)^T A (x_t - x^*) = \frac{1}{2}[A^{-1} g_t]^T A [A^{-1} g_t] = \frac{1}{2} g_t^T A^{-1} g_t.$$

Combining the resulting relations, we come to

$$\frac{E(x_t) - E(x_{t+1})}{E(x_t)} = \frac{(g_t^T g)^2}{[g_t^T A g_t][g_t^T A^{-1} g_t]},$$

as required in (6.1.39).

the coordinates on the plane). Consequently, (u, v) belongs to the convex hull C of Γ. This convex hull is as you see on the picture:



Arc Γ and its convex hull

The largest, over $(u, v) \in C$, product uv corresponds to the case when (u, v) belongs to the line segment $[P_1, P_n]$ bounding C from above, so that

$$uv \leq \max_{0\leq a\leq 1}[(a + (1 - a)Q)(a + \frac{1 - a}{Q})];$$

the right hand side maximum can be explicitly computed (it corresponds to $a = 1/2$), and the resulting value is $(Q + 1)^2/(4Q)$, as claimed. ■

Combining Lemma 6.1.2 and (6.1.39), we come to the following

Proposition 6.1.6 [Convergence ratio for the Steepest Descent as applied to strongly convex quadratic form]
As applied to a strongly convex quadratic form f with condition number Q, the Steepest Descent converges linearly with the convergence ratio not worse than

$$1 - \frac{4Q}{(Q + 1)^2} = \left(\frac{Q - 1}{Q + 1}\right)^2, \tag{6.1.41}$$

namely, for all N one has

$$[f(x_{N+1}) - \min f] \leq \left(\frac{Q - 1}{Q + 1}\right)^2 [f(x_N) - \min f],$$

whence also

$$f(x_N) - \min f \leq \left(\frac{Q - 1}{Q + 1}\right)^{2N} [f(x_0) - \min f]. \tag{6.1.42}$$

Note that the Proposition says that the convergence ratio is *not worse* than $(Q-1)^2(Q+1)^{-2}$; the actual convergence ratio depends on the starting point x_0. It is known, anyhow, that (6.1.42) gives correct description of the rate of convergence: for "almost all" starting points, the process indeed converges at the rate close to the indicated upper bound. Since the convergence ratio given by Proposition is $1 - O(1/Q)$ (cf. (6.1.32)), quantitative conclusion (**) from the previous subsection indeed is valid, even in the case of strongly convex *quadratic* f.

Local rate of convergence of Steepest Descent. Relation (6.1.42) is a *non-asymptotical* efficiency estimate of the Steepest Descent in the *quadratic* case. In the *non-quadratic nondegenerate* case the method admits similar *asymptotic* efficiency estimate. Namely, one can prove the following

Theorem 6.1.2 [Local rate of convergence of Steepest Descent]
Assume that the trajectory $\{x_t\}$ of the Steepest Descent as applied to f converges to a point x^* which is a nondegenerate local minimizer of f, namely, is such that f is twice continuously differentiable in a neighbourhood of x^* and the Hessian $\nabla^2 f(x^*)$ of the objective at x^* is positive definite.

Then the trajectory converges to x^* linearly, and the convergence ratio of the sequence $f(x_t) - f(x^*)$ of residuals in terms of the objective is not worse than

$$\left(\frac{Q-1}{Q+1}\right)^2,$$

Q being the condition number of $\nabla^2 f(x^*)$:

$$(\forall \epsilon > 0\ \exists C_\epsilon < \infty): \quad f(x_N) - f(x^*) \le C_\epsilon \left(\frac{Q-1}{Q+1} + \epsilon\right)^{2N},\ N = 1, 2, ... \tag{6.1.43}$$

6.1.5 Conclusions

Let us summarize our knowledge of Gradient Descent. We know that

- In the most general case, under mild regularity assumptions, both StD and ArD converge to the set of critical points of the objective (see Theorem 6.1.1), and there is certain guaranteed sublinear rate of global convergence in terms of the quantities $|\nabla f(x_N)|^2$ (see Proposition 6.1.1);
- In the convex $C^{1,1}$ case ArD converges to a global minimizer of the objective (provided that such a minimizer exists), and there is certain guaranteed (sublinear) rate of global convergence in terms of the residuals in the objective $f(x_N) - \min f$ (see Proposition 6.1.2);
- In the strongly convex case ArD converges to the unique minimizer of the objective, and both distances to the minimizer and the residuals in terms of the objective admit global linearly converging to zero upper bounds. The corresponding convergence ratio is given by the condition number of the objective Q (see Proposition 6.1.5) and is of the type $1 - O(1/Q)$, so that the number of steps required to reduce the initial inaccuracy by a given factor is proportional to Q (this is an upper bound, but typically it reflects the actual behaviour of the method);
- It the quadratic case - globally, and in the nonquadratic one – asymptotically, StD converges linearly with the convergence ratio $1 - O(1/Q)$, Q being the condition number of the Hessian of the objective at the minimizer towards which the method converges (in the quadratic case, of course, this Hessian simply is the matrix of our quadratic form).

This is what we know. What should be conclusions – is the method good, or bad, or what? As it normally is the case in numerical optimization, we are unable to give a definite answer: there are too many different criteria to be taken into account. What we can do, is to list advantages and disadvantages of the method. Such a knowledge provides us with a kind of orientation: when we know what are the strong and the weak points of an optimization method and given a particular application we are interested in, we can decide "how strong in the case in question are the strong points and how weak are the weak ones", thus getting possibility to choose the solution method better fitting the situation. As about the Gradient Descent, the evident strong points of the method are

- broad family of problems where we can guarantee global convergence to a critical point (normally - to a local minimizer) of the objective;
- simplicity: at a step of the method, we need single evaluation of ∇f and a small number of evaluations of f (the evaluations of f are required by the line search; if one uses ArD with simplified line search mentioned in Section 5.2.4.1, this number indeed is small). Note that each evaluation of f is accompanied by small ($O(n)$, n being the dimension of the design vector) number of arithmetic operations.

The most important weak point of the method is relatively low rate of convergence: even in the strongly convex quadratic case, the method converges linearly. This itself is not that bad; what indeed is bad, is that the convergence ratio is too sensitive to the condition number Q of the objective. As we remember, the number of steps of the method, for a given progress in accuracy, is proportional to Q. And this indeed is too bad, since in applications we typically meet with ill-conditioned problems, with condition numbers of orders of thousands and millions; whenever this is the case, we hardly can expect something good from Gradient Descent, at least when we are interested in high-accuracy solutions.

It is worthy to understand what is the geometry underlying slowing down the Gradient Descent in the case of ill-conditioned objective. Consider the case of strongly convex quadratic f. The level surfaces

$$S_\delta = \{x \mid f(x) = \min f + \delta\}$$

of f are homothetic ellipsoids centered at the minimizer x^* of f; the squared half-axes of these ellipsoids are inverse proportional to the eigenvalues of $A = \nabla^2 f$. Indeed, as we know from (6.1.35),

$$f(x) = \frac{1}{2}(x - x^*)^T A(x - x^*) + \min f,$$

so that in the orthogonal coordinates x_i associated with the orthonormal eigenbasis of A and the origin placed at x^* we have

$$f(x) = \frac{1}{2}\sum_i \lambda_i x_i^2 + \min f,$$

λ_i being the eigenvalues of A. Consequently, the equation of S_δ in the indicated coordinates is

$$\sum_i \lambda_i x_i^2 = 2\delta.$$

Now, if A is ill-conditioned, the ellipsoids S_δ become a kind of "valleys" – they are relatively narrow in some directions (those associated with smallest half-axes of the ellipsoids) and relatively long in other directions (associated with the largest half-axes). The gradient – which is orthogonal to the level surface – on the dominating part of this surface looks "almost across the valley", and since the valley is narrow, the steps turn out to be short. As a result, the trajectory of the method is a kind of short-step zig-zag movement with slow overall trend towards the minimizer.

What should be stressed is that in the case in question there is nothing intrinsically bad in the problem itself; all difficulties come from the fact that we relate the objective to a "badly chosen" initial coordinates. Under appropriate *non-orthogonal* linear transformation of coordinates (pass from x_i to $y_i = \sqrt{\lambda_i} x_i$) the objective becomes perfectly conditioned – it becomes the sum of squares of the coordinates, so that condition number now equals 1, and the Gradient Descent, *being run in the new coordinates*, will go directly towards the minimizer. The problem, of course, is that the Gradient Descent is *associated with a once for ever fixed initial Euclidean coordinates* (since the underlying notion of gradient is a Euclidean notion: different Euclidean structures result in different gradient vectors of the same function at the same point). If these initial coordinates are badly chosen for a given objective f (so that the condition number of f with respect to these coordinates is large), the Gradient Descent will be slow, although if we were clever enough to perform first appropriate *scaling* – linear non-orthogonal transformation of the coordinates – and then run Gradient Descent in these new coordinates, we might obtain fast convergence. In the next Section we will consider the famous *Newton method* which, in a sense, is nothing but "locally optimally scaled" Gradient Descent, with the scaling varying from step to step.

6.2 Basic Newton's Method

We continue investigating methods for unconstrained minimization problem

$$f(x) \to \min \mid x \in \mathbf{R}^n.$$

What is on our agenda is the famous *Newton method* based on local quadratic model of f. To get possibility to speak about this model, we assume from now on that f is twice continuously differentiable.

6.2.1 The Method

The idea of the method is very simple; we have already used this idea in the univariate case (Lecture 8). Given current iterate x, the value $f(x)$, the gradient $\nabla f(x)$ and the Hessian $\nabla^2 f(x)$ of the objective at x, we approximate f around x by its second order Taylor expansion

$$f(y) \approx f(x) + (y-x)^T \nabla f(x) + \frac{1}{2}(y-x)^T [\nabla^2 f(x)](y-x)$$

and take as the next iterate the minimizer of the right hand side quadratic form of y. To get this minimizer, we differentiate the right hand side in y and set the gradient to 0, which results in the equation with respect to y

$$[\nabla^2 f(x)](y-x) = -\nabla f(x).$$

This is a linear system with respect to y; assuming the matrix of the system (i.e., the Hessian $\nabla^2 f(x)$) nonsingular, we can write down the solution as

$$y = x - [\nabla^2 f(x)]^{-1} \nabla f(x).$$

In the Basic Newton method, we simply iterate the indicated updating:

Algorithm 6.2.1 [Basic Newton Method] *Given starting point x_0, run the recurrence*

$$x_t = x_{t-1} - [\nabla^2 f(x_{t-1})]^{-1} \nabla f(x_{t-1}). \tag{6.2.1}$$

The indicated method is not necessarily well-defined (e.g., what to do when the Hessian at the current iterate turns out to be singular?) We shall address this difficulty, same as several other difficulties which may occur in the method, in the next Lecture. Our current goal is to establish the fundamental result on the method – its *local quadratic convergence in the non-degenerate case:*

Theorem 6.2.1 [Local Quadratic Convergence of the Newton method in the nondegenerate case] *Assume that f is three times continuously differentiable in a neighbourhood of $x^* \in \mathbf{R}^n$, and that x^* is a nondegenerate local minimizer of f, i.e., $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then the Basic Newton method, starting close enough to x^*, converges to x^* quadratically.*

Proof. Let U be a convex neighbourhood of x^* where the third order partial derivatives of f (i.e., the second order partial derivatives of the components of ∇f) are bounded. In this neighbourhood, consequently,

$$|-\nabla f(y) - \nabla^2 f(y)(x^* - y)| \equiv |\nabla f(x^*) - \nabla f(y) - \nabla^2 f(y)(x^* - y)| \leq \beta_1 |y - x^*|^2 \tag{6.2.2}$$

with some constant β_1 (we have applied to the components of ∇f the standard upper bound on the remainder in the first order Taylor expansion: if $g(\cdot)$ is a scalar function with bounded second order derivatives in U, then

$$|g(x) - g(y) - \nabla g(y)(x-y)| \leq \beta |y-x|^2$$

for some $\beta < \infty$ [2)] and all $x, y \in U$, cf. Lemma 3.3.1).

Since $\nabla^2 f(x^*)$ is nonsingular and $\nabla^2 f(x)$ is continuous at $x = x^*$, there exists a smaller neighbourhood $U' \subset U$ of x^*, let it be the centered at x^* ball of radius $r > 0$, such that

$$y \in U' \Rightarrow |[\nabla^2 f(y)]^{-1} \leq \beta_2 \tag{6.2.3}$$

for some constant $beta_2$; here and in what follows, for a matrix A $|A|$ denotes the *operator norm* of A, i.e.,

$$|A| = \max_{|h| \leq 1} |Ah|,$$

[2] note that the magnitude of β is of order of the magnitude of second order derivatives of g in U

the right hand side norms being the standard Euclidean norms on the corresponding vector spaces.

Now assume that certain point x_t of the trajectory of the Basic Newton method on f is close enough to x^*, namely, is such that

$$x_t \in U'', \quad U'' = \{x \mid |x - x^*| \leq \rho \equiv \min[\frac{1}{2\beta_1\beta_2}, r]\}. \tag{6.2.4}$$

We have

$$|x_{t+1} - x^*| = |x_t - x^* - [\nabla^2 f(x_t)]^{-1}\nabla f(x_t)| =$$

$$= |[\nabla^2 f(x_t)]^{-1}\left[\nabla^2 f(x_t)(x_t - x^*) - \nabla f(x_t)\right]| \leq |[\nabla^2 f(x_t)]^{-1}||-\nabla f(x_t) - \nabla^2 f(x_t)(x^* - x_t)| \leq$$

[by (6.2.3) and (6.2.2)]

$$\leq \beta_1\beta_2|x_t - x^*|^2.$$

Thus, we come to

$$x_t \in U'' \Rightarrow |x_{t+1} - x^*| \leq \beta_1\beta_2|x_t - x^*|^2 \quad [\leq (\beta_1\beta_2|x_t - x^*|)|x_t - x^*| \leq 0.5|x_t - x^*|]. \tag{6.2.5}$$

We see that the new iterate x_{t+1} is at least twice closer to x^* than x_t and, consequently, $x_{t+1} \in U''$. Thus, reaching U'' at certain moment $\bar{t}$ (this for sure happens when the trajectory is started in U''), the trajectory never leaves this neighbourhood of x^*, and

$$|x_{t+1} - x^*| \leq \beta_1\beta_2|x_t - x^*|^2 \leq 0.5|x_t - x^*|, \ t \geq \bar{t},$$

so that the trajectory converges to x^* quadratically. ■

The indicated theorem establishes fast – quadratic – local convergence of the Basic Newton method to a nondegenerate local minimizer of f, which is fine. At the same time, we remember from Lecture 8 that even in the univariate case and for smooth convex objective, the Newton method not necessarily possesses global convergence: started not too close to minimizer, it may diverge. It follows that we cannot rely on this method "as it is" – in actual computations, how could we know that the starting point is "close enough" to the minimizer? Thus, some modifications are needed in order to make the method globally converging. Right now we shall look at the simplest modification of this type, postponing more serious modifications till the next Lecture.

6.2.2 Incorporating line search

The Basic Newton method at each iteration performs a unit step:

$$x_{t+1} = x_t = e(x_t)$$

in the *Newton direction*

$$e(x) = -[\nabla^2 f(x)]^{-1}\nabla f(x). \tag{6.2.6}$$

It prescribes, consequently, both the search direction and the stepsize (just 1) in the direction. The first idea how to "cure" the method to make it globally convergent is to use only the direction given by the method, but not the stepsize; as about the stepsize, we could choose it by a kind of line search aimed to achieve "significant progress" in the objective (compare with the Gradient Descent). Thus, we come to the *Newton method with line search* given by the recurrence

$$x_{t+1} = x_t + \gamma_{t+1}e(x_t), \ e(x_t) = [\nabla^2 f(x_t)]^{-1}\nabla f(x_t), \tag{6.2.7}$$

where the stepsize $\gamma_{t+1} \geq 0$ is given by a line search. In particular, we could speak about

- *"Steepest Newton method"*:

$$\gamma_{t+1} \in \mathrm{Argmin}\{f(x_t + \gamma e(x_t)) \mid \gamma \geq 0\},$$

or

- the *Armijo-based Newton method*, where γ_{t+1} is given by the Armijo-terminated line search,

  or

- the *Goldstein-based Newton method*, where γ_{t+1} is given by the Goldstein-terminated line search.

We could expect the indicated modifications to make the method globally converging; at the same time, we may hope that close enough to a nondegenerate local minimizer of f, the indicated line search will result in stepsize close to 1, so that the asymptotical behaviour of the modified method will be similar to the one of the basic method (provided, of course, that the modified method indeed converges to a nondegenerate local minimizer of f). Whether these hopes indeed are valid or not, it depends on f, and at least one property of f seems to be necessary to make our hopes valid: $e(x)$ should be a *descent* direction of f at a non-critical x:

$$\nabla f(x) \neq 0 \Rightarrow e^T(x)\nabla f(x) \equiv -(\nabla f(x))^T[\nabla^2 f(x)]^{-1}\nabla f(x) < 0. \tag{6.2.8}$$

Indeed, if there exists x with nonzero $\nabla f(x)$ such that the Newton direction $e(x)$ is not a descent direction of f at x, it is unclear whether there exists a step in this direction which reduces f. If, e.g., $f(x+\gamma e(x))$ is a nondecreasing function of $\gamma > 0$, then the Steepest Newton method started at x clearly will never leave the point, thus converging to (simply staying at) non-critical point of f. Similar difficulties occur with the Armijo- and Goldstein-based Newton methods: is $e(x)$ is not a descent direction of f at x, then the Armijo/Goldstein-terminated line search makes no sense at all.

We see that one may hope to prove convergence of the Newton method with line search only if f satisfies the property

$$\nabla f(x) \neq 0 \Rightarrow (\nabla f(x))^T[\nabla^2 f(x)]^{-1}\nabla f(x) > 0.$$

The simplest way to impose this property is to assume that f is convex with nonsingular Hessian:

$$\nabla^2 f(x) > 0 \quad \forall x \quad \left[\equiv h^T[\nabla^2 f(x)]h > 0 \ \forall x \forall h \neq 0\right]; \tag{6.2.9}$$

indeed, if the matrix $\nabla f(x)$ is positive definite for every x, then, as it is known from Linear Algebra (and can be proved immediately), the matrix $[\nabla^2 f(x)]^{-1}$ also is positive definite at every x, so that (6.2.8) takes place. It indeed turns out that under assumption (6.2.9) the line search versions of the Newton method possess global convergence:

Proposition 6.2.1 *Let f be a twice continuously differentiable convex function with the Hessian $\nabla^2 f(x)$ being positive definite at every point x, and let x_0 be such that the level set*

$$S = \{x \mid f(x) \leq f(x_0)\}$$

associated with x_0 is bounded. Then the Steepest Newton method, same as the Armijo/Goldstein-based Newton method, started at x_0 converges to the unique global minimizer of f.

The proof of the Proposition is completely similar to the one of Theorem 6.1.1 and is therefore omitted.

The "line search" modification of the Basic Newton method is not quite appropriate: as we just have seen, in this modification we meet with severe difficulties when the Newton direction at certain point is not a descent direction of the objective. Another difficulty is that the Newton direction, generally speaking, can simply be undefined – $\nabla^2 f(x)$ may be singular at a point of the trajectory. What to do in this situation? We see that in order to make the Newton method reliable, we need to modify not only the stepsize used in the method, but also the search direction itself, at least in the cases when it is "bad" (is not a descent direction of f or simply is undefined). We shall discuss the corresponding modifications later.

6.2.3 The Newton Method: how good it is?

We have investigated the basic version of the Newton method, along with its modifications aimed to make the method globally converging in the convex case. Finally, what are the basic advantages and disadvantages of the method?

The main advantage of the method is its fast (quadratic) local convergence to a nondegenerate local minimizer of the objective, provided that we were lucky to bring the trajectory close enough to such a minimizer.

Rigorously speaking, the indicated attractive property is possessed only by the basic version of the method. For a modified version, the indicated phenomenon takes place only when the modified method manages to drive the trajectory to a small neighbourhood of a nondegenerate local minimizer of f; if it is not so, we have no reasons to expect fast convergence. Thus, let us assume that *the trajectory of the modified Newton method converges to a nondegenerate local minimizer x^* of f*[3)]. Is then the convergence asymptotically quadratic?

The answer depends on the rules for line search; indeed, to get something asymptotically close to the Basic Newton method, we need nearly unit stepsizes at the final phase of the process. One can prove, e.g., that the required property of "asymptotically unit stepsizes in the Newton direction" is ensured by the exact line search. To get the same behaviour in the Armijo-terminated line search, the parameters ϵ and η of the underlying Armijo test should be chosen properly (e.g., $\epsilon = 0.2$ and $\eta = 10$), and we should always start the line search with testing the unit stepsize.

In spite of all indicated remarks which say that the modifications of the Basic Newton method aimed to ensure global convergence may spoil the theoretical quadratic convergence of the basic method (either because of bad implementation, or due to degeneracy of the minimizer the trajectory converges to), the Newton-based methods should be qualified as the most efficient tool for smooth unconstrained minimization. The actual reason of the efficiency of these methods is their intrinsic ability (spoiled, to some extent, by the modifications aimed to ensure global convergence) to adjust themselves to the "local geometry" of the objective.

The main shortcoming of the Newton-type methods is their relatively high computational cost. To run such a method, we should be able to compute the Hessian of the objective and should solve at each step an $n \times n$ system of linear equations. If the objective is too complicated and/or the dimension n of the problem is large, these requirements may become "too costly" from the viewpoint of coding, execution time and memory considerations. In order to overcome these drawbacks, significant effort was invested into theoretical and computational development of first-order routines (those not using second-order derivatives) capable to "imitate" the Newton method. These are the methods we are about to consider in Lecture 10.

6.2.4 Newton Method and Self-Concordant Functions

The traditional results on the Newton method establish no more than its fast *asymptotical* convergence; global efficiency estimates can be proved only for modified versions of the method, and these global estimates never are better than those for the Gradient Descent. In this section we consider a family of objectives – the so called *self-concordant ones* – where the Newton method admits excellent *global* efficiency estimates. This family underlies the most recent and advanced *Interior Point methods for large-scale Convex Optimization*.

6.2.4.1 Preliminaries

The traditional "starting point" in the theory of the Newton method – Theorem 6.2.1 – possesses an evident drawback (which, anyhow, remained unnoticed by generations of researchers). The Theorem establishes local quadratic convergence of the Basic Newton method as applied to a function f with positive definite Hessian at the minimizer, this is fine; but what is the "quantitative" information given by the Theorem? What indeed is the "region of quadratic convergence" $\mathcal{Q}$ of the method – the set of those starting points from which the method converges quickly to x^*? The proof provides us with certain "constructive" description of $\mathcal{Q}$, but look – this description involves differential characteristics of f like

[3)] this, e.g., for sure is the case when f is strongly convex: here the only critical point is a nondegenerate global minimizer, while convergence to the set of critical points is given by Proposition 6.2.1

the magnitude of the third order derivatives of f in a neighbourhood of x^* (this quantity underlies the constant β_1 from the proof) and the bound on the norm of inverted Hessian in this neighbourhood (the constant β_2; in fact this constant depends on β_1, the radius of the neighbourhood and the smallest eigenvalue of $\nabla^2 f(x^*)$). Besides this, the "fast convergence" of the method is described in terms of the behaviour of the standard Euclidean distances $|x_t - x^*|$. All these quantities – magnitudes of third-order derivatives of f, norms of the inverted Hessian, distances from the iterates to the minimizer – are "frame-dependent": they depend on the choice of Euclidean structure on the space of variables, on what are the orthonormal coordinates used to compute partial derivatives, Hessian matrices and their norms, etc. When we vary the Euclidean structure (pass from the original coordinates to another coordinates via a non-orthogonal linear transformation), all these quantities somehow vary, same as the description of $\mathcal{Q}$ given by Theorem 6.2.1. On the other hand, when passing from one Euclidean structure on the space of variables to another, we do not vary neither the problem, nor the Basic Newton method. Indeed, the latter method is independent of any a priori coordinates, as it is seen from the following "coordinateless" description of the method:

To find the Newton iterate x_{t+1} of the previous iterate x_t, take the second order Taylor expansion of f at x_t and choose, as x_{t+1}, the minimizer of the resulting quadratic form.

Thus, the coordinates are responsible only for the *point of view* we use to investigate the process and are absolutely irrelevant to the process itself. And the results of Theorem 6.2.1 in their quantitative part (same as other traditional results on the Newton method) reflect this "point of view", not only the actual properties of the Newton process! This "dependence on viewpoint" is a severe drawback: how can we get correct impression of actual abilities of the method looking at the method from an "occasionally chosen" position? This is exactly the same as to try to get a good picture of a landscape directing the camera in a random manner.

6.2.4.2 Self-concordance

After the drawback of the traditional results is realized, could we choose a proper point of view – to orient our camera properly, at least for "good" objectives? Assume, e.g., that our objective f is convex with nondegenerate Hessian. Then at every point x there is a natural, intrinsic for the objective, Euclidean structure on the space of variables, namely, the one given by the Hessian of the objective at x; the corresponding norm is

$$|h|_{f,x} = \sqrt{h^T \nabla^2 f(x) h} \equiv \sqrt{\frac{d^2}{dt^2}|_{t=0} f(x+th)}. \tag{6.2.10}$$

Note that the first expression for $|h|_{f,x}$ seems to be "frame-dependent" – it is given in terms of coordinates used to compute inner product and the Hessian. But in fact the value of this expression is "frame-independent", as it is seen from the second representation of $|h|_{f,x}$.

Now, from the standard results on the Newton method we know that the behaviour of the method depends on the magnitudes of the third-order derivatives of f. Thus, these results are expressed, among other, in terms of upper bounds

$$\left| \frac{d^3}{dt^3}|_{t=0} f(x+th) \right| \leq \alpha$$

on the third-order directional derivatives of the objective, *the derivatives being taken along unit in the standard Euclidean metric directions h.* What happens if we impose similar upper bound on the third-order directional derivatives *along the directions of the unit $|\cdot|_{f,x}$ length*

rather than along the directions of the unit "usual" length? In other words, what happens if we assume that

$$|h|_{f,x} \leq 1 \Rightarrow \left|\frac{d^3}{dt^3}|_{t=0} f(x+th)\right| \leq \alpha \quad ?$$

Since the left hand side of the concluding inequality is of homogeneity degree 3 with respect to h, the indicated assumption is equivalent to the one

$$\left|\frac{d^3}{dt^3}|_{t=0} f(x+th)\right| \leq \alpha |h|^3_{f,x} \quad \forall x \; \forall h.$$

Now, the resulting inequality, qualitatively, remains true when we *scale* f – replace it by λf with positive constant λ, but the value of α varies: $\alpha \mapsto \lambda^{-1/2}\alpha$. We can use this property to normalize the constant factor α, e.g., to set it equal to 2 (this is the most technically convenient normalization).

Thus, we come to the main ingredient of the notion of a

self-concordant function: a three times continuously differentiable convex function f satisfying the inequality

$$\left|\frac{d^3}{dt^3}|_{t=0} f(x+th)\right| \leq 2|h|^3_{f,x} \equiv 2\left[\frac{d^2}{dt^2}|_{t=0} f(x+th)\right]^{d3/2} \quad \forall h \in \mathbf{R}^n. \tag{6.2.11}$$

We do not insist on f to be defined everywhere; it suffices to assume that the domain of f is an open convex set $Q_f \subset \mathbf{R}^n$, and that (6.2.11) is satisfied at every point $x \in Q_f$. The second part of the definition of a self-concordant function is that

Q_f is a "natural domain" of f, so that f possesses the barrier property with respect to Q_f – blows up to infinity when a sequence of interior points of Q_f approaches a boundary point of the domain:

$$\forall \{x_i \in Q_f\}: \; x_i \to x \in \partial Q_f, i \to \infty \Rightarrow f(x_i) \to \infty, \; i \to \infty. \tag{6.2.12}$$

Of course, the second part of the definition imposes something on f only when the domain of f is less than the entire $\mathbf{R}^n$.

Note that the definition of a self-concordant function is "coordinateless" – it imposes certain inequality between third- and second-order directional derivatives of the function and certain behaviour of the function on the boundary of its domain; all notions involved are "frame-independent".

6.2.4.3 Self-concordant functions and the Newton method

It turns out that the Newton method *as applied to a self-concordant function* f possesses extremely nice *global* convergence properties. Namely, one can more or less straightforwardly prove the following statements:

Proposition 6.2.2 [Self-concordant functions and the Newton method]

Let f be strongly self-concordant, and let $\nabla^2 f$ be nondegenerate at some point of Q_f (this for sure is the case when Q_f does not contain lines, e.g., is bounded). Then

- (i) [Nondegeneracy] *$\nabla f(x)$ is positive definite at every point $x \in Q_f$;*
- (ii) [Existence of minimizer] *If f is below bounded (which for sure is the case when Q_f is bounded), then f attains its minimum on Q_f, the minimizer being unique;*

- (iii) [Damped Newton method] *The started at arbitrary point $x_0 \in Q_f$ process*

$$x_{t+1} = x_t - \frac{1}{1+\lambda(f,x_t)}[\nabla^2 f(x_t)]^{-1}\nabla f(x_t), \quad \lambda(f,x) = \sqrt{(\nabla f(x))^T[\nabla^2 f(x)]^{-1}\nabla f(x)} \tag{6.2.13}$$

  – *the Newton method with particular stepsizes*

$$\gamma_{t+1} = \frac{1}{1+\lambda(f,x_t)}$$

  – *possesses the following properties:*
    – (iii.1) *The process keeps the iterates in Q_f and is therefore well-defined;*
    – (iii.2) *If f is below bounded on Q_f (which for sure is the case if $\lambda(f,x) < 1$ for some $x \in Q_f$) then $\{x_t\}$ converges to the unique minimizer x_f^* of f on Q_f;*
    – (iii.3) *Each step of the process (6.2.13) decreases f "significantly", provided that $\lambda(f,x_t)$ is not too small:*

$$f(x_t) - f(x_{t+1}) \geq \lambda(f,x_t) - \ln(1+\lambda(f,x_t)); \tag{6.2.14}$$

    – (iii.4) *For every t, one has*

$$\lambda(f,x_t) < 1 \Rightarrow f(x_t) - f(x_t^*) \leq -\ln(1-\lambda(f,x_t)) - \lambda(f,x_t) \tag{6.2.15}$$

  and

$$\lambda(f,x_{t+1}) \leq \frac{2\lambda^2(f,x_t)}{1-\lambda(f,x_t)}. \tag{6.2.16}$$

The indicated statements demonstrate extremely nice *global convergence* properties of the Damped Newton method (6.2.13) as applied to a self-concordant function f. Namely, assume that f is self-concordant with nondegenerate Hessian at certain (and then, as it was mentioned in the above proposition, at any) point of Q_f. Assume, besides this, that f is below bounded on Q_f (and, consequently, attains its minimum on Q_f by (ii)). According to (iii), the Damped Newton method is keeps the iterates in A_f. Now, we may partition the trajectory into two parts:

- the initial phase: from the beginning to the first moment, let it be called t^*, when $\lambda(f,x_t) \leq 1/4$;
- the final phase: starting from the moment t^*.

According to (iii.3), at every step of the initial phase the objective is decreased at least by *absolute* constant

$$\kappa = \frac{1}{4} - \ln\frac{5}{4} > 0;$$

consequently,

- *the initial phase is finite and is comprised of no more than*

$$\mathcal{N}_{\text{ini}} = \frac{f(x_0) - \min_{Q_f} f}{\kappa}$$

  iterations.

Starting with $t = t^*$, we have in view of (6.2.15):

$$\lambda(f,x_{t+1}) \leq \frac{2\lambda^2(f,x_t)}{1-\lambda(f,x_t)} \leq \frac{1}{2}\lambda(f,x_t);$$

thus,

- *starting with $t = t^*$, the quantities $\lambda(f, x_t)$ converge quadratically to 0 with objective-independent rate.*

  According to (6.2.16),

- *starting with $t = t^*$, the residuals in terms of the objective $f(x_t) - \min_{Q_f} f$ also converge quadratically to zero with objective-independent rate.*

Combining the above observations, we observe that

- *the number of steps of the Damped Newton method required to reduce the residual $f(x_t) - \min f$ in the value of a self-concordant below bounded objective to a prescribed value $\epsilon < 0.1$ is no more than*

$$N(\epsilon) \leq O(1)\left[[f(x_0) - \min f] + \ln\ln\frac{1}{\epsilon}\right], \tag{6.2.17}$$

  $O(1)$ being an absolute constant.

It is also worthy of note what happens when we apply the Damped Newton method to a *below unbounded* self-concordant f. The answer is as follows:

- *for a below unbounded f one has $\lambda(f, x) \geq 1$ for every x (see (iii.2)), and, consequently, every step of the method decreases f at least by the absolute constant $1 - \ln 2$ (see (iii.3)).*

The indicated picture gives a "frame-" and "objective-independent" description of the *global* behaviour of the Damped Newton method as applied to a below bounded self-concordant function. Note that the quantity $\lambda(f, x)$ used to describe the behaviour of the method at the first glance is "coordinate dependent" (see (6.2.13)), but in fact this quantity is "coordinateless". Indeed, one can easily verify that

$$\frac{\lambda^2(f, x)}{2} = \hat{f}(x) - \min_y \hat{f}(y),$$

where

$$\hat{f}(y) = f(x) + (y - x)^T \nabla f(x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$

is the second-order Taylor expansion of f at x. This is a coordinateless definition of $\lambda(f, x)$.

Note that the region of quadratic convergence of the Damped Newton method as applied to a below bounded self-concordant function f is, according to (iii.4), the set

$$\mathcal{Q}_f = \{x \in Q_f \mid \lambda(f, x) \leq \frac{1}{4}\}. \tag{6.2.18}$$

6.2.4.4 Self-concordant functions: applications

At the first glance, the family of self-concordant functions is rather "thin" – the functions are given by certain "strict" differential inequality, and "a general", even convex and smooth, f hardly may happen to be self-concordant. Thus, what for these elegant results on the behaviour of the Newton method on self-concordant functions?

The answer is as follows: it turns out that (even constrained) Convex Programming problems of reasonable (in principle – of arbitrary) analytic structure can be reduced to a "small series" of problems of minimizing self-concordant functions. Applying to these auxiliary problems the Damped Newton method, we come to the theoretically most efficient (and extremely efficient in practice) *Interior Point Polynomial Time methods for Convex Optimization*. Appearance of these methods (starting with the landmark paper of N. Karmarkar (1984), where the first method of this type for Linear Programming was proposed) definitely

was the main event in Optimization during the last decade, it completely changed the entire area of large-scale Convex Optimization, in particular, Linear Programming.

Right now we are not going to speak about Interior Point methods in more details; we shall come back to these methods at the end of our course. What should be stressed now is that *the crucial point in the design of the Interior Point methods is our ability to construct "good" self-concordant functions with prescribed domains*. To this end it is worthy to note how to construct self-concordant functions. Here the following "raw materials" and "combination rules" are useful:

Raw materials: basic examples of self-concordant functions. For the time being, the following examples are sufficient:

- [Convex quadratic (e.g., linear) form] The convex quadratic function

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c$$

  (A is symmetric positive semidefinite $n \times n$ matrix) is self-concordant on $\mathbf{R}^n$;

- [Logarithm] The function

$$-\ln(x)$$

  is self-concordant with the domain $\mathbf{R}_+ = \{x \in \mathbf{R} \mid x > 0\}$;

- [Extension of the previous example: Logarithmic barrier, linear/quadratic case] Let

$$Q = \{x \in \mathbf{R}^n \mid \phi_j(x) < 0,\ j = 1, ..., m\}$$

  be a nonempty set in $\mathbf{R}^n$ given by m strict convex quadratic (e.g., linear) inequalities. Then the function

$$f(x) = -\sum_{i=1}^{m} \ln(-\phi_i(x))$$

  is self-concordant with the domain equal to Q.

Combination rules: simple operations with functions preserving self-concordance

- [Linear combination with coefficients ≥ 1] Let f_i, $i = 1, ...m$, be self-concordant functions with the domains Q_{f_i}, let these domains possess a nonempty intersection Q, and let $\alpha_i \geq 1$, $i = 1, ..., m$, be given reals. Then the function

$$f(x) = \sum_{i=1}^{m} \alpha_i f_i(x)$$

  is self-concordant with the domain equal to Q.

  In particular, the sum of a self-concordant function and a convex quadratic function (e.g., a linear one) is self-concordant;

- [Affine substitution] Let $f(x)$ be self-concordant with the domain $Q_f \subset \mathbf{R}^n$, and let $x = A\xi + b$ be an affine mapping from $\mathbf{R}^k$ into $\mathbf{R}^n$ with the image intersecting Q_f. Then the composite function

$$g(\xi) = f(A\xi + b)$$

  is self-concordant with the domain

$$Q_g = \{\xi \mid A\xi + b \in Q_f\}$$

  being the inverse image of Q_f under the affine mapping in question.

To justify self-concordance of the indicated functions, same as the validity of the combination rules, only minimal effort is required; at the same time, these examples and rules give almost all required to establish excellent global efficiency estimates for Interior Point methods as applied to Linear Programming and Convex Quadratically Constrained Quadratic programming.

After we know examples of self-concordant functions, let us look how our now understanding of the behaviour of the Newton method on such a function differs from the one given by Theorem 6.2.1. To this end consider a particular self-concordant function – the logarithmic barrier

$$f(x) = -\ln(\delta - x_1) - \ln(\delta + x_1) - \ln(1 - x_2) - \ln(1 + x_2)$$

for the 2D rectangle

$$D = \{x \in \mathbf{R}^2 \mid |x_1| < \delta, |x_2| < 1\};$$

in what follows we assume that the rectangle is "wide", i.e., that

$$\delta >> 1.$$

This function indeed is self-concordant (see the third of the above "raw material" examples). The minimizer of the function clearly is the origin; the region of quadratic convergence of the Damped Newton method is given by

$$\mathcal{Q} = \{x \in D \mid \frac{x_1^2}{\delta^2 + x_1^2} + \frac{x_2^2}{1 + x_2^2} \leq \frac{1}{32}\}$$

(see (6.2.18)). We see that the region of quadratic convergence of the Damped Newton method is large enough – it contains, e.g., 8 times smaller than D concentric to D rectangle D'. Besides this, (6.2.17) says that in order to minimize f to inaccuracy, in terms of the objective, ϵ, starting with a point $x_0 \in D$, it suffices to perform no more than

$$O(1)\left[\ln \frac{1}{\| x_0 \|} + \ln \ln \frac{1}{\epsilon}\right]$$

steps, where $O(1)$ is an absolute constant and

$$\| x \| = \max\{\frac{|x_1|}{\delta}, |x_2|\}.$$

Now let us look what Theorem 6.2.1 says. The Hessian $\nabla^2 f(0)$ of the objective at the minimizer is

$$H = \begin{pmatrix} 2\delta^{-2} & 0 \\ 0 & 2 \end{pmatrix},$$

and $|H^{-1}| = O(\delta^2)$; in, say, 0.5-neighbourhood U of $x^* = 0$ we also have $|[\nabla^2 f(x)]^{-1}| = O(\delta^2)$. The third-order derivatives of f in U are of order of 1. Thus, in the notation from the proof of Theorem 6.2.1 we have $\beta_1 = O(1)$ (this is the magnitude of the third order derivatives of f in U), $U' = U$, $r = 0.5$ (the radius of the circle $U' = U$) and $\beta_2 = O(\delta^2)$ (this is the upper bound on the norm of the inverted Hessian of f in U'). According to the proof, the region U'' of quadratic convergence of the Newton method is ρ-neighbourhood of $x^* = 0$ with

$$\rho = \min[r, (2\beta_1\beta_2)^{-1}] = O(\delta^{-2}).$$

Thus, according to Theorem 6.2.1, the region of quadratic convergence of the method becomes the smaller the larger is δ, while the actual behaviour of this region is quite opposite.

In this simple example, the aforementioned drawback of the traditional approach – its "frame-dependence" – is clearly seen. Applying Theorem 6.2.1 to the situation in question, we used extremely bad "frame" – Euclidean structure. If we were clever enough to scale the variable

x_1 before applying Theorem 6.2.1 – to divide it by δ – it would become absolutely clear that the behaviour of the Newton method is absolutely independent of δ, and the region of quadratic convergence of the method is a once for ever fixed "fraction" of the rectangle D.

To extract certain moral from this story about self-concordance, let me note that it is one of the examples of what is Mathematics and progress in Mathematics: the known from XVII Century very natural and simple (and seemingly perfectly well understood decades ago) optimization method gave rise to one of the most advanced and most recent breakthroughs in Optimization.

Lecture 7

Around the Newton Method

In the previous Lecture we spoke about the Newton method

$$x_{t+1} = x_t + \gamma_{t+1} e_t, \quad e_t = -[\nabla^2 f(x_t)]^{-1} \nabla f(x_t),$$

f being the objective to be minimized; from now on we assume the objective to be smooth enough (namely, 3 times continuously differentiable) and such that the level set

$$S = \{x \mid f(x) \leq f(x_0)\},$$

x_0 being the starting point of the method in question, is bounded.

We have considered two versions of the method – the *basic* one, where we all the time use the unit stepsizes ($\gamma_{t+1} \equiv 1$), and version with line search, where γ_{t+1} is given by a kind of line search applied to the function

$$\phi(\gamma) = f(x_t + \gamma e_t)$$

– to the restriction of the objective on the "search ray" given by the current iterate x_t and the "Newton direction" e_t.

As we remember, the Basic version of the method, being started close enough to a nondegenerate local minimizer x^* of f (one with positive definite $\nabla^2 f(x^*)$) converges to x^* quadratically, although may diverge if started far from the optimal set. The Line search version of the method possesses global convergence, provided that the objective is convex with nondegenerate Hessian ($\nabla^2 f(x)$ is positive definite for every x). The latter assumption is, however, too restrictive for many applications, and this fact motivates the first of our current goals:

> **A.** *To modify the Newton method in order to make it globally convergent independently of the convexity and nondegeneracy assumptions*

Of course, the only kind of convergence it makes sense to bother about in the general (nonconvex) case is the convergence to the set of critical points of the objective, not to the set of global minimizers of it. And of course the modifications we are interested in should preserve the most attractive property of the Basic Newton method – its fast local convergence: if the method converges to a nondegenerate local minimizer of f, the convergence should, at least asymptotically, be as fast as for the Basic method.

A. is the first of our goals, but not the only one of them. It was mentioned in Lecture 9 that from the practical viewpoint a severe shortcoming of the Newton method is the necessity to compute the Hessians of the objective and to invert these Hessians. This shortcoming motivates the second of our goals:

> **B.** *To "imitate" the behaviour of the Newton method in a first-order scheme (one using only the first order derivatives of f and not inverting matrices)*

We shall overview four generic modifications of the Newton method:

- *Newton Method with Cubic Regularization*

- *Modified Newton Methods*
- *Conjugate Gradient Methods*
- *Quasi-Newton Methods*

The Newton Method with Cubic Regularization and the Modified Newton methods take care of the goal **A** only and ignore the goal **B** – they still use the second order derivatives and invert matrices. In contrast to this, both Conjugate Gradient and Quasi-Newton methods are aimed to achieve both goals **A** and **B**; the difference between these two families is in *how* they try to achieve these goals.

7.1 Newton Method with Cubic Regularization

Chronologically, Newton Method with Cubic Regularization is the latest addition the "Newton Toolbox" – it was proposed by Yu. Nesterov and B. Polyak in 2006 [1], an instructive example of a genuine breakthrough in the area which for many years seemed to be fully understood. We shall present this method in the simplest case of the problem

$$f(x) \to \min | x \in \mathbf{R}^n \tag{7.1.1}$$

of unconstrained minimization of three times continuously differentiable function f. The idea of the method is very simple. Assume for starters that the third derivative of f is bounded, the third directional derivative of f, taken at any point along any direction of unit length does not exceed some constant $L \in (0, \infty)$ depending solely on f. Then for every x and h it holds

$$f(x+h) \leq \overline{f}_x(h) := f(x) + h^T \nabla f(x) + \frac{1}{2} h^T \nabla^2 f(x) h + \frac{L}{6} \parallel h \parallel^3 . \tag{7.1.2}$$

(Theorem A.6.5). Now, the right hand side in this inequality as a function of h goes to $+\infty$ as $\parallel h \parallel \to \infty$ and as such achieves minimum over $h \in \mathbf{R}^n$, let a (global) minimizer be h_*. Observe that $\overline{f}_x(h_*) \leq \overline{f}_x(0)$, and this inequality is strict, unless $h = 0$ is a global minimizer of $\overline{f}_x$. Assuming that the latter is not the case, we have

$$f(x+h_*) \leq \overline{f}_x(h_*) < \overline{f}_x(0) = f(x), \tag{7.1.3}$$

where the first inequality is given by (7.1.2). Thus, in the case in question passing from x to $x^+ = x + h_*$, we strictly reduce the value of f. On the other hand, when $h = 0$ is a global minimizer of $\overline{f}_x(h)$, we should have $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$ (by second order necessary optimality conditions for unconstrained minimization as applied to $\overline{f}_x(\cdot)$; note that this function is twice continuously differentiable). Iterating the construction, we arrive at the algorithm

$$x_{t+1} = x_t + h_t, \ h_t \in \operatorname*{Argmin}_h [f(x_t) + h^T \nabla f(x_t) + \frac{1}{2} h^T \nabla^2 f(x_t) h + \frac{L}{6} \parallel h \parallel^3] \tag{7.1.4}$$

From the above analysis it follows that the algorithm is well-defined and monotone:

$$f(x_{t+1}) < f(x_t)$$

for all t such that x_t does *not* satisfy the second order necessary optimality conditions. If x_t does satisfy these conditions, then $\overline{f}_t(\cdot)$ is a convex function satisfying $\overline{f}_{x_t}(h) > \overline{f}_{x_t}(0)$ whenever $h \neq 0$, implying that $h = 0$ is the only minimizer of the function; consequently, in the case in question the algorithm gets stuck at x_t: $x_t = x_{t+1} = x_{t+2} = ...$

So far, we have assumed that f is three times continuously differentiable with the third derivative bounded on the entire $\mathbf{R}^n$; besides this, we have implicitly assumed that we know constant L participating in (7.1.2) – this constant is used in the recurrence (7.1.4). Both these assumptions can be weakened, specifically,

[1] Yu. Nesterov and B.T. Polyak, Cubic regularization of Newton method and its global performance, *Mathematical Programming* **108** (2006), 177-205.

1. We can assume that $f : \mathbf{R}^n \to \mathbf{R}$ is three times continuously differentiable on a closed and *bounded* convex set X which is "large enough," specifically, int X contains the level set
$$X_0 = \{x \in X : f(x) \le f(x_0)\},$$
where $x_0 \in \text{int } X$ is a given starting point;

2. Instead of recurrence (7.1.4), we can use the recurrence
$$x_{t+1} = x_t + h_t,\ h_t \in \operatorname*{Argmin}_h [f(x_t) + h^T \nabla f(x_t) + \frac{1}{2} h^T \nabla^2 f(x_t) h + \frac{L_t}{6} \| h \|^3], \tag{7.1.5}$$
where $L_t > 0$ is *good*, goodness meaning that the point x_{t+1} defined according to (7.1.5) belongs to X and satisfies the inequality
$$f(x_{t+1}) \le f(x_t) + h^T \nabla f(x_t) + \frac{1}{2} h^T \nabla^2 f(x_t) h + \frac{L_t}{6} \| h \|^3; \tag{7.1.6}$$
whenever this is the case, we, same as above, have
$$x_{t+1} \in X_0 \ \&\ f(x_{t+1}) \le \min_h [f(x_t) + h^T \nabla f(x_t) + \frac{1}{2} h^T \nabla^2 f(x_t) h + \frac{L_t}{6} \| h \|^3] \le f(x_t), \tag{7.1.7}$$
where the concluding inequality is strict, provided x_t does not satisfy second order necessary conditions for (unconstrained) local optimality.

It is immediately seen that when $x_t \in X_0$, all large enough values of L_t are good; specifically, we have the following

Lemma 7.1.1 *Let*
$$M = M_X(f) = \max_{x \in X, h \in \mathbf{R}^n, \|h\| \le 1} |\frac{d^3}{dt^3}|_{t=0} f(x + th)|.$$
Whenever $x_t \in X_0$, every $L_t \ge M$ is good.

Proof. To save notation, and w.l.o.g., let $x_t = 0$, let $L \ge M$, and let $\bar{h}$ be a minimizer of the function $\phi(h) := f(0) + h^T \nabla f(0) + \frac{1}{2} h^T \nabla^2 f(0) h + \frac{L}{6} \| h \|^3$. Observe that $\bar{h}^T \nabla f(0) \le 0$, since otherwise we would have $\phi(-\bar{h}) < \phi(\bar{h})$. Observe that setting $\gamma(t) = \phi(t\bar{h})$, $t \ge 0$, we get a cubic polynomial with $\gamma'(0) = \bar{h}^T \nabla f(0) \le 0$ and $\gamma'(1) = 0$; besides this, $\gamma'(t)$ is a convex quadratic function. As a result, we have $\gamma'(t) \le 0$, $0 \le t \le 1$, implying that $\gamma(t) \le \gamma(0) = f(0)$ for $t \in [0, 1]$. Now let $\bar{t}$ be the largest $t \in [0, 1]$ such that $t\bar{h} \in X$, so that the function $g(t) = f(t\bar{h})$ is well defined on $[0, \bar{t}]$, and since $L \ge M$, we have $g(t) \le \gamma(t)$ for $0 \le t \le \bar{t}$, which combines with the facts that $x_t = 0 \in X_0$ and $\gamma(t) \le f(0)$, $0 \le t \le 1$, to imply that $t\bar{h} \in X_0$ for $t \le \bar{t}$. Recalling what $\bar{t}$ is and that $X_0 \subset \text{int } X$, we conclude that $\bar{t} = 1$ and that
$$f(\bar{h}) = g(1) \le \gamma(1) = \phi(\bar{h}) = \min_h [f(0) + h^T \nabla f(0) + \frac{1}{2} h^T \nabla^2 f(0) h + \frac{L}{6} \| h \|^3],$$
which combines with the already established fact that $\bar{h} = \bar{t}x \in X_0$ to imply that $L_t = L$ is good for $x_t = 0$. ∎

What we would like to use, is the "nearly smallest" good value of L_t (since the less is L_t, the less is the upper bound on $f(x_{t+1})$ given by (7.1.7)). Such a value can be rapidly found by a kind of "line search," namely, as follows. Given x_t and L_{t-1} (where L_{-1} is, say, 1), we check whether $L_t = L_{t-1}$ is good; if it is the case, we test for goodness one by one the values $L_{t-1}/2$, $L_{t-1}/4$, $L_{t-1}/8$,... until either the value of L_t we are testing looses goodness, or it becomes less than a small a priori selected threshold, say, 10^{-6}; in both cases, the last good value of L_t we have observed is used to build the actual, and not just a candidate, next iterate x_{t+1} according to (7.1.5). On the other hand, when the value L_{t-1} of L_t turns out to be bad, we test one by one the values $2L_{t-1}$, $4L_{t-1}$, $8L_{t-1}$,..., until the first good value of L_t is met, and use this value to build the next iterate. Note that with this policy, the values of L_t used to build x_{t+1} are bounded by $2\max[M_X(f), L_{-1}]$.

It is easily seen that given X and x_0 as above, the just described algorithm with line search is well defined and monotone:

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq ... \tag{7.1.8}$$

with all inequalities being strict, unless the algorithms "gets stuck" at a point where the necessary second order optimality conditions are met. It is not difficult to verify that if the algorithms does *not* get stuck, every limiting point of the trajectory (the trajectory is bounded by (7.1.8); recall that X_0 is bounded) satisfies the second order necessary optimality conditions for unconstrained minimization. Moreover, given $\epsilon > 0$, $\delta > 0$, one can explicitly upper-bound, via ϵ, δ, $M_X(f)$ and the diameter of X, the number of steps t in which the second order necessary optimality conditions are satisfied within accuracy (ϵ, δ), that is, x_t satisfies the relation

$$\| \nabla f(x_t) \| \leq \epsilon, \lambda_{\min}(\nabla^2 f(x_t)) \geq -\delta,$$

where $\lambda_{\min}(A)$ is the minimal eigenvalue of a symmetric matrix A. Moreover, it can be proved (for the proofs of all announced results, see the cited paper of Nesterov and Polyak) that if among the limiting points of the trajectory there is a point $\bar{x}$ where the necessary second order conditions for local optimality are satisfied (i.e., $\nabla f(\bar{x}) = 0$, $\nabla^2 f(\bar{x}) \succ 0$), then in fact the trajectory converges to $\bar{x}$ quadratically.

Pay attention to the extremely attractive property of the Newton method with cubic regularization – under mild smoothness and boundedness assumptions, all limiting points of the trajectory provably satisfy *second order* necessary optimality conditions, which is in sharp contrast with basically all other traditional algorithms for smooth unconstrained minimization – for these algorithms, one can prove only that all limiting points of the trajectory are critical points of the objective, those where the first order optimality conditions for unconstrained minimization are satisfied.

7.1.1 Implementing the algorithm

The question we did not answer so far is how to carry out an iteration of the algorithm, that is, how to find global minimizer of a function of the form

$$\phi(h) = a^T h + h^T Q h + \rho \| h \|^3,$$

where $\rho > 0$ and Q is a symmetric matrix. The simplest, to the best of our knowledge, answer to this question is given by the following algorithm:

1. We start with computing eigenvalue decomposition $Q = UMU^T$ (U is orthogonal, $M = \mathrm{Diag}\{\mu_1, ..., \mu_n\}$ is diagonal) of Q. Passing from the original variable h to the variable $y = U^T h$, we reduce the problem of minimizing $\phi(\cdot)$ to the problem of minimizing in y the function

$$\psi(y) = b^T y + \sum_{i=1}^n \mu_i y_i^2 + \rho(\sum_i y_i^2)^{3/2} \qquad [b = U^T a]$$

2. It is clear that at a global minimizer of ψ, the signs of y_i are opposite to the signs of b_i, that is, $b_i y_i \leq 0$ (otherwise replacing y_i with $-y_i$, we would reduce the value of the function). Thus, to minimize $\psi(y)$ is the same as to minimize the function

$$\omega(z) = -\sum_i |b_i| z_i + \sum_i \mu_i z_i^2 + \rho(\sum_i y_i^2)^{3/2}$$

over $z \geq 0$: a minimizer z^* of the latter function gives rise to the global minimizer y^* of ψ given by $y_i^* = -\mathrm{sign}(b_i) z_i^*$.

3. Finally, minimizing $\omega(z)$ over $z \geq 0$ via substitution of variables $z_i = \sqrt{\zeta_i}$ reduces to minimizing function

$$\chi(\zeta) = \sum_i \left[\mu_i \zeta_i - |b_i| \sqrt{\zeta_i}\right] + \rho(\sum_i \zeta_i)^{3/2}$$

over the nonnegative orthant, which is easy – χ is convex!.

The simplest way to minimize χ over $\zeta \geq 0$ is to rewrite the problem as

$$\min_{\zeta,r}\left[\sum_i[\mu_i\zeta_i - \beta_i\sqrt{\zeta_i}] + \rho r^{3/2}\right] \mid 0 \leq \zeta_i, \sum_i \zeta_i \leq r \qquad [\beta_i = |b_i|]$$

and pass to the Lagrange function

$$\begin{array}{rcl} L(\zeta, r; \lambda) & = & \sum_i[\mu_i\zeta_i - \beta_i\sqrt{\zeta_i}] + \rho r^{3/2} + \lambda(\sum_i \zeta_i - r) \\ & = & \sum_i[(\mu_i + \lambda)\zeta_i - \beta_i\sqrt{\zeta_i}] + [\rho r^{3/2} - \lambda r]. \end{array}$$

Given $\lambda \geq 0$, we can explicitly minimize the Lagrange function over $\zeta \geq 0, r \geq 0$; assuming $\beta_i > 0$ rather than $\beta_i \geq 0$ (which always can be achieved by arbitrarily small perturbation of β_i's), the corresponding minimizer $\zeta(\lambda), r(\lambda)$ is unique and continuously depends on λ. As a result, we can rapidly solve by Bisection the Lagrange dual

$$\max_{\lambda \geq 0}\left[\overline{L}(\lambda) = \min_{\zeta \geq 0, r \geq 0} L(\zeta, r; \lambda)\right].$$

of our problem of interest; after a high accuracy solution λ_* to the dual problem is found, we can recover a high accuracy minimizer of χ as $\zeta_* = \zeta(\lambda_*)$.

Note that the most expensive computationally step in the algorithm – eigenvalue decomposition of A - does not operate with ρ. As a result, in the above "line search" implementation of the Newton method with cubic regularization, adjusting L_t is relatively cheap (the eigenvalue decomposition of $\nabla^2 f(x_t)$ should be done once, and the remaining effort is dominated by the necessity to compute the values of f at the candidates to the role of x_{t+1} stemming from the trial values of L_t).

7.2 Modified Newton methods

7.2.1 Variable Metric Methods

Let us start with developing certain general approach which covers both the Gradient and the Newton methods and allows to understand how to "cure" the Newton method. To outline the idea, let us come back to the Gradient Descent. What in fact we did in this method? Given previous iterate x, we used local *linear* approximation of the objective:

$$f(y) \approx f(x) + (y - x)^T \nabla f(x) \qquad (7.2.1)$$

and choose, as the next direction of movement, the "most perspective" of descent directions of the right hand side. Now, how we were comparing the directions to choose the "most perspective" one? We took the unit ball

$$W = \{d \mid d^T d \leq 1\}$$

of directions and choose in this ball the direction which minimizes the value

$$\bar{f}(x + d) \equiv f(x) + d^T \nabla f(x)$$

of the approximate objective. This direction, as it is immediately seen, is simply the normalized anti-gradient direction

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

and in the Gradient Descent we used it as the current direction of movement, choosing the stepsize in order to achieve "significant" progress in the objective value. Note that instead of minimizing $\bar{f}(x + d)$ on the ball W, we could minimize the quadratic function

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

over $d \in \mathbf{R}^n$; the result will be simply the anti-gradient direction $-\nabla f(x)$. This is not the same as the above normalized anti-gradient direction, but the difference in normalization is absolutely unimportant

for us – in any case we indent to use line search in the generated direction, so that what in fact we are interested in is the *search ray* $\{x + \gamma d \mid \gamma \geq 0\}$, and proportional, with positive coefficient, directions result in the same ray.

With the outlined interpretation of the Gradient Descent as a method with line search and the search direction given by minimization of the linearized objective $\bar{f}(x + d)$ over $d \in W$, we may ask ourselves: why we use in this scheme the unit ball W, not something else? For example, why not to use an ellipsoid

$$W_A = \{d \mid d^T A d \leq 1\},$$

A being a positive definite symmetric matrix?
Recall that an ellipsoid in properly chosen coordinates of $\mathbf{R}^n$ becomes a ball (the coordinates are obtained from the standard ones by linear nonsingular transformation). Consequently, a method of the outlined type with W replaced by W_A, i.e., the one where the search direction is given by

$$d = \operatorname*{argmin}_{d \in W_A} \bar{f}(x + d) \tag{7.2.2}$$

has the same "right to exist" as the Gradient Descent. This new "scaled by matrix A Gradient Descent" is nothing but the usual Gradient Descent, but associated with the coordinates on $\mathbf{R}^n$ where W_A becomes the unit ball. Note that the usual Gradient Descent corresponds to the case $A = I$, I being the unit matrix. Now, the initial coordinates are absolutely "occasional" – they have nothing in common with the problem we are solving; consequently, we have no reason to prefer these particular coordinates. Moreover, if we were lucky to adjust the coordinates we use to the "geometry" of the objective (cf. discussion in the previous Lecture), we could get a method with better convergence than the one of the usual Gradient Descent.

Same as above, the direction given by (7.2.2) is, up to renormalization (the latter, as it was already explained, is unimportant – it is "suppressed" by line search), nothing but the direction given by the minimization of the quadratic form

$$\hat{f}_A(d) = d^T \nabla f(x) + \frac{1}{2} d^T A d; \tag{7.2.3}$$

minimizing the right hand side with respect to d (to this end it suffices to solve the Fermat equation $\nabla_d \hat{f}_A(d) \equiv \nabla f + Ad = 0$), we come to the explicit form of the search direction:

$$d = -A^{-1} \nabla f(x). \tag{7.2.4}$$

Note that this direction for sure is a descent direction of f at x, provided that x is not a critical point of f:

$$\nabla f(x) \neq 0 \Rightarrow d^T \nabla f(x) = -(\nabla f(x))^T A^{-1} \nabla f(x) < 0$$

(recall that A is symmetric positive definite, whence A^{-1} also is symmetric positive definite), so that we are in a good position to apply to f line search in the direction d.

The summary of our considerations is as follows: choosing a positive definite symmetric matrix A, we can associate with it "A-anti-gradient direction" $-A^{-1}\nabla f(x)$, which is a descent direction of f at x (provided that $\nabla f(x) \neq 0$). And we have the same reasons to use this direction in order to improve f as those to use the standard anti-gradient direction (given by the same construction with $A = I$).

Now we can make one step of generalization more: why should we use at each step of the method a once for ever fixed matrix A instead of varying this matrix from iteration to iteration? The "geometry" of the objective varies along the trajectory, and it is natural to adjust the matrix A to this varying geometry. Thus, we come to the following generic scheme of a *Variable Metric method:*

Algorithm 7.2.1 [Generic Variable Metric method]
Initialization: choose somehow starting point x_0 and set $t = 0$
Step t: given previous iterate x_{t-1},

- *compute $f(x_{t-1})$, $\nabla f(x_{t-1})$ and, possibly, $\nabla^2 f(x_{t-1})$;*

- *choose somehow positive definite symmetric matrix A_t and compute the A_t-anti-gradient direction*

$$d_t = -A_t^{-1}\nabla f(x_{t-1})$$

of f at x_{t-1};

- *perform line search from x_{t-1} in the direction d_t, thus getting new iterate*

$$x_t = x_{t-1} + \gamma_t d_t \equiv x_{t-1} - \gamma_t A_t^{-1}\nabla f(x_{t-1}),$$

replace t with $t+1$ and loop.

The outlined scheme covers all methods we know so far: to get different versions of the Gradient Descent, we should set $A_t \equiv I$ and should specify the version of the line search to be used. With $A_t = \nabla^2 f(x_{t-1})$, we get, as d_t, the Newton direction of f at x_{t-1}, and we come to the Newton method with line search; further specifying the line search by the "programmed" rule $\gamma_t = 1$, we get the Basic Newton method. Thus, *the Basic Newton method is nothing but the Gradient Descent scaled by the Hessian of the objective at the current iterate*. Note, anyhow, that the "Newton choice" $A_t = \nabla^2 f(x_{t-1})$ is compatible with the outlined scheme (where A_t should be symmetric positive definite) only when $\nabla^2 f(x_{t-1})$ is positive definite[2)]; from the above discussion we know that if it is not the case, we indeed have no reasons to use the Newton direction and should somehow modify it to make it descent. Thus, the generic Algorithm 7.2.1 covers all we know to the moment and provides us with good idea how to "cure" the Newton method at a "bad" iterate:

> *at such an iterate, we should replace the actual Hessian $A_t = \nabla^2 f(x_{t-1})$ in the expression $-A_t^{-1}\nabla f(x_{t-1})$ for the Newton direction by its positive definite "correction" in order to make the resulting direction descent for the objective.*

Thus, we come to the family of *Modified Newton methods* – those given by the generic Algorithm 7.2.1 where we use as A_t, whenever it is possible, the Hessian $\nabla^2 f(x_{t-1})$ of the objective, and when it is impossible, choose as A_t a "positive definite correction" of the current Hessian.

Before passing to the Modified Newton methods themselves, we should understand whether our new scheme indeed achieves our target **A** – whether it makes the modified method globally converging.

7.2.2 Global convergence of a Variable Metric method

We are about to prove that the generic Variable Metric method (Algorithm 7.2.1), under some reasonable restrictions on the matrices A_t, globally converges to the set of critical points of f. The restrictions are very simple: in the generic method, A_t should be symmetric positive definite, and what we need is "uniform" positive definiteness.

Let us say that Algorithm 7.2.1 is *uniformly descent* with parameters p, P, $0 < p \le P < \infty$, if the rules for choosing A_t in the algorithm ensure that

$$\lambda_{\min}(A_t) \ge p, \quad \lambda_{\max}(A_t) \le P, \quad t = 1, 2, ... \tag{7.2.5}$$

(as always, $\lambda_{\min}(A)$and $\lambda_{\max}(A)$ denote the minimal and the maximal eigenvalues of a symmetric matrix A). Thus, "uniform descentness" simply means that the matrices A_t never become "too large" (their maximal eigenvalues are bounded away from infinity) and never become "almost degenerate" (their minimal eigenvalues are bounded away from zero).

Theorem 7.2.1 [Global convergence of uniformly descent Variable Metric method]
Let f be twice continuously differentiable function, and let x_0 be such that the level set

$$S = \{x \in \mathbf{R}^n \mid f(x) \le f(x_0)\}$$

[2)] the scheme requires also symmetry of A_t, but here we have no problems: since f is from the very beginning assumed to be three times continuously differentiable, its Hessians for sure are symmetric

is bounded. Assume that f is minimized by a uniformly descent Variable Metric method started at x_0, and assume that the line search used in the method is either the exact one-dimensional line search, or the Armijo-terminated one. Then the trajectory of the method is bounded, the objective is non-increasing along the trajectory, and all limiting points of the trajectory (which for sure exist, since the trajectory is bounded) belong to the set

$$X^{**} = \{x \mid \nabla f(x) = 0\}$$

of critical points of f.

Proof is similar to the one of Theorem 6.1.1 and is therefore omitted. It is an excellent (non-obligatory!) exercise to restore the proof and to understand what is the role of the "uniform descentness". ■

7.2.3 Implementations of the Modified Newton method

Now let us come back to the Newton method. As we remember, our goal is to modify it in order to assure global convergence (at least the same as the one for the Gradient Descent); and, of course, we would not like to loose the fine local convergence properties of the method given by Theorem 6.2.1. We already have an idea how to modify the method: we should use line search in a descent direction given by the Variable Metric scheme, where we use, as A_t, the Hessian $\nabla^2 f(x_{t-1})$, if the latter matrix is positive definite; if it is not the case, we use as A_t certain positive definite correction of the Hessian. Of course, from both theoretical and numerical reasons, we should modify $\nabla^2 f(x_{t-1})$ not only when it fails to be positive definite, but also when it is "close" to a non-positive-definite matrix. Indeed, if we do not modify $\nabla^2 f(x_{t-1})$ when it is close to a non-positive-definite matrix, i.e., when the minimal eigenvalue of the Hessian, being positive, is small, we are unable to ensure the uniform descent property of the method and, consequently, are unable to ensure global convergence; this is a theoretical reason. The practical one is that in the outlined case the condition number of the Hessian is large, so that when solving numerically the Newton system

$$\nabla^2 f(x_{t-1})e = -\nabla f(x_{t-1})$$

in order to find the Newton direction, we meet with severe numerical problems, and the actually computed direction will be far from the exact one.

Thus, our general tactics in the Newton-based implementation of Algorithm 7.2.1 should be as follows: given x_{t-1}, we compute

$$H_{t-1} \equiv \nabla^2 f(x_{t-1})$$

and check whether this matrix is "well positive definite"; if it is the case, we use this matrix as A_t, otherwise replace H_{t-1} with certain "well positive definite" correction A_t of H_{t-1}. Let us look at two simplest implementations of the outlined scheme.

7.2.3.1 Modifications based on Spectral Decomposition

Recall that, as it is known from Linear Algebra, any symmetric $n \times n$ matrix A possesses *spectral decomposition*

$$A = UDU^T,$$

U being an $n \times n$ orthogonal matrix:

$$U^T U = I,$$

and D being a diagonal matrix. The diagonal entries of D are exactly the eigenvalues of A, while the columns of U are the associated normalized eigenvectors. In particular, A is positive definite if and only if the diagonal entries of D are positive.

In the Modified Newton methods based on spectral decomposition one finds the indicated decomposition of the Hessian at every step:

$$H_t \equiv \nabla^2 f(x_t) = U_t D_t U_t^T, \tag{7.2.6}$$

and compares the eigenvalues of H_t – i.e., the diagonal entries of D_t – with a once for ever chosen "threshold" $\delta > 0$. If all the eigenvalues are $\geq \delta$, we qualify H_t as "well positive definite" and use it as A_{t+1}. If some of the eigenvalues of H_t are $< \delta$ (e.g., are negative), we set

$$A_{t+1} = U_t \bar{D}_t U_t^T,$$

where $\bar{D}_t$ is the diagonal matrix obtained from D by replacing the diagonal entries smaller than δ by δ. Another way to "correct" H_t is to replace the negative diagonal values in D_t by their absolute values (and then to replace by δ those diagonal entries, if any, which are less than δ).

Both indicated strategies result in

$$\lambda_{\min}(A_t) \geq \delta, \ t = 1, 2, ...,$$

and never increase "significantly" the norm of the Hessian:

$$\lambda_{\max}(A_t) \leq \max\left[|\lambda_{\max}(H_t)|, |\lambda_{\min}(H_t)|, \delta\right];$$

as a result, the associated modified Newton method turns out to be uniformly descent (and thus globally converging), provided that the level set of f associated with the starting point is bounded (so that the Hessians along the trajectory are uniformly bounded). A drawback of the approach is its relatively large computational cost: to find spectral decomposition (7.2.6) to machine precision, it normally requires between $2n^3$ and $4n^3$ arithmetic operations, n being the row size of H_t (i.e., the design dimension of the optimization problem). As we shall see in a while, this is, in a sense, too much.

7.2.3.2 Levenberg-Marquardt Modification

In the *Levenberg-Marquardt* modification of the Newton method we choose A_{t+1} as the "regularization"

$$A_{t+1} = \epsilon_t I + H_t \tag{7.2.7}$$

of the actual Hessian, where the "regularization parameter" $\epsilon_t \geq 0$ is chosen to make the right hand side "well positive definite" – to have all its eigenvalues at least the chosen in advance positive threshold δ. This is the same as to ensure that $A_{t+1} \geq \delta I$ [3)].

To find the desired ϵ_t, we first check whether the requirement

$$H_t > \delta I$$

is satisfied; if it is the case, we choose $\epsilon_t = 0$ and $A_{t+1} = H_t$, thus getting the pure Newton direction. Now assume that the inequality $H_t \geq \delta I$ does not take place. The matrix

$$A(\epsilon) = \epsilon I + H_t - \delta I$$

for sure is positive definite when $\epsilon > 0$ is large enough, and in the case in question it is not positive definite for $\epsilon = 0$. Thus, there exists the smallest $\epsilon = \epsilon^* \geq 0$ for which $A(\epsilon)$ is positive semidefinite. And what we do in the Levenberg-Marquardt scheme are several steps of the bisection routine to find a "tight" upper bound ϵ_t for ϵ^*. The resulting upper bound is used to create A_t according to (7.2.7).

The essence of the matter is, of course, how we verify whether a given trial value of ϵ is appropriate, i.e., whether it results in positive semidefinite $A(\epsilon)$. It would be absolutely senseless to answer this question by computing spectral decomposition of $A(\epsilon)$ – this computation is exactly what we are trying to avoid. Fortunately, in the Linear Algebra there is a much more efficient algorithm for checking positive definiteness of given symmetric matrix – the Choleski factorization. This factorization answers the question in approximately $n^3/6$ arithmetic operations, n being the size of the matrix we are factorizing. In the Levenberg-Marquardt scheme, we use this Choleski factorization to govern the Bisection; with this approach, one can check positive definiteness of $A(\epsilon)$ for a given ϵ in about $n^3/6$ arithmetic operations – something from 12 to 24 times cheaper than the cost of spectral decomposition. Thus, the Levenberg-Marquardt scheme with 5-10 bisection steps in ϵ (this is sufficient for "smart" implementation of the scheme) is numerically less expensive then the scheme based on spectral decomposition.

[3)] from now on, for symmetric matrices A, B the inequality $A \geq B$ means that $A - B$ is positive semidefinite, and $A > B$ means that $A - B$ is positive definite. Note that, for a symmetric matrix A, the relation $\lambda_{\max}(A) \leq a$ is equivalent to $aI - A \geq 0$, while $\lambda_{\min}(A) \geq a$ is equivalent to $A - aI \geq 0$

7.2.3.3 Choleski Factorization

It is known from Linear Algebra, that a symmetric $n \times n$ matrix A is positive definite is and only if it admits factorization

$$A = LDL^T, \tag{7.2.8}$$

where

- L is lower-triangular $n \times n$ matrix with unit diagonal entries;
- D is diagonal matrix with positive diagonal entries.

The Choleski Factorization is an algorithm which computes the factors L and D in decomposition (7.2.8), if such a decomposition exists. The algorithm is given by the recurrence

$$d_j = a_{ii} - \sum_{s=1}^{j-1} d_s l_{js}^2, \tag{7.2.9}$$

$$l_{ij} = \frac{1}{d_j}\left(a_{ij} - \sum_{s=1}^{j-1} d_s l_{js} l_{is}\right), \quad j \le i \le n, \tag{7.2.10}$$

(d_j is j-th diagonal entry of D, l_{ij} and a_{ij} are the entries of L and A, respectively). The indicated recurrence allows to compute D and L, if they exist, in

$$\mathcal{C}_n = \frac{n^3}{6}(1 + o(1))$$

arithmetic operations ($o(1) \to 0$, $n \to \infty$), in a numerically stable manner. Note that L is computed in the "column by column" fashion: the order of computations is

$$d_1 \to l_{1,1}, l_{d2,1}, ..., l_{n,1} \to d_2 \to l_{d2,2}, l_{d3,2}, ..., l_{n,2} \to d_3 \to l_{d3,3}, l_{d4,3}, ...l_{n,3} \to ... \to d_n \to l_{n,n};$$

if the right hand side in (7.2.9) turns out to be nonpositive for some j, this indicates that A is not positive definite.

The main advantage of Choleski factorization is not only its ability to check in $\mathcal{C}_n$ computations whether A is positive definite, but also to get, as a byproduct, the factorization (7.2.8) of a positive definite A. With this factorization, we can immediately solve the linear system

$$Ax = b;$$

the solution x to the system can be identified by backsubstitutions – sequential solving, for u, v and x, two triangular and one diagonal systems

$$Lu = b; \quad Dv = u; \quad L^T x = v,$$

and this computation requires only $O(n^2)$ arithmetic operations. The resulting method – Choleski decomposition with subsequent backsubstitutions (the *square root method*) – is thought to be the most efficient in the operation count and most numerically stable Linear Algebra routine for solving linear systems with general-type symmetric positive definite matrices.

7.3 Conjugate Gradient Methods

The family of methods in question address both our goals **A** and **B**. The idea of the methods is as follows. What we are doing in the Newton method can be explained as follows: we believe that our objective f

is a quadratic form, and apply direct Linear Algebra tools to find, under this assumption, its minimizer – namely, form and solve with respect to x the linear system

$$(N) \quad \nabla^2 f(x_t)(x - x_t) = -\nabla f(x_t),$$

which, for the case of quadratic (and convex) f would give us the minimizer of f. In the non-quadratic case the resulting x, of course, does not minimize f, and we treat it as our new iterate. The drawbacks of the scheme we are trying to eliminate are that we need second-order information on f to assemble the Newton system (N) and should solve the system in order to get x.

In the Conjugate Gradient scheme we also act as if we were believing that the objective is quadratic, but instead of direct forming and solving (N) we solve the system by an iterative method. It turns out that one can choose this method in such a way that it

- (i) does not involve explicitly the matrix of the system; all operations are described in terms of the values and the first order derivatives of f at subsequent iterates;
- (ii) solves the system exactly in no more than n steps, n being the dimension of x.

Since the iterative method for solving (N) we use is described in terms of the values and the gradients of f, it *formally* can be applied to an arbitrary smooth objective; and if the objective turns out to be convex quadratic form, the method, by (ii), will find its minimizer in at most n steps. In view of the latter property of the method we could expect that if the objective is "nearly quadratic", then n steps of the method, although not resulting in exact minimizer of f, give us a much better approximation to the minimizer than the point the method is started from. Choosing this approximation as our new starting point and performing n steps of the method more, we may hope for new significant reduction in inaccuracy, and so on. Of course, all these hopes are under the assumption that the objective is "nearly quadratic"; but this indeed will be eventually our case, if the method will converge, and this latter property will indeed be ensured by our construction.

It is clear from the outlined general idea that the "main ingredient" of the scheme is certain iterative method for minimizing convex quadratic forms, and we start with the description of this method.

7.3.1 Conjugate Gradient Method: Quadratic Case

Let

$$f(x) = \frac{1}{2} x^T H x - b^T x \tag{7.3.1}$$

be a strongly convex quadratic form, so that H is positive definite symmetric matrix. The unique global minimizer x^* of f is the solution to the Fermat equation

$$\nabla f(x) \equiv Hx - b = 0; \tag{7.3.2}$$

due to strong convexity, this is the only critical point of f. Thus, it is the same – to minimize a strongly convex quadratic form f on $\mathbf{R}^n$ and to solve a linear $n \times n$ system of equations

$$Hx = b \tag{7.3.3}$$

with symmetric positive definite matrix $H = \nabla^2 f$.

We already spoke about direct Linear Algebra technique for solving (7.3.3) – namely, about the one based on Choleski decomposition of H. There are other direct Linear Algebra methods to solve this problem – Gauss elimination, etc. When saying that these methods are "direct", I mean that they work with H "as a whole", and until the matrix is processed, no approximate minimizers of f (i.e., approximate solutions to (7.3.3)) are generated; and after H is processed, we get the exact minimizer (in actual computations – exact up to influence of rounding errors). In contrast to this, the so called *iterative* methods for minimizing f ($\equiv$ for solving (7.3.3)) generate a sequence of approximate minimizers converging to the exact one. We already know a method of this type – the Steepest Descent; this method is aimed to minimize nonquadratic functions, but one can apply it to a quadratic objective as well.

Now we shall speak about another iterative method for minimizing f – the *Conjugate Gradient* one. I shall start with a *non-iterative* description of the method.

7.3.1.1 CG: Initial description

Let x_0 be the starting point we use when minimizing our quadratic objective. We associate with this point the *Krylov vectors*

$$g_0 \equiv Hx_0 - b, Hg_0, H^2g_0, H^3g_0, ...$$

and the *Krylov subspaces*

$$E_0 = \{0\},\ E_t = \mathrm{Lin}\{g_0, Hg_0, H^2g_0, ..., H^{t-1}g_0\},\ t = 1, 2, ...$$

so that E_t is the linear span of the first t Krylov vectors. It is easily seen that

- The Krylov subspaces grow with t:

$$E_1 \subset E_2 \subset E_3 \subset ...$$

- Let $k \geq 0$ be the first value of t such that the first t Krylov vectors are linearly dependent. Then the inclusion

$$E_t \subset E_{t+1}$$

is strict for $t < k - 1$ and is equality for $t \geq k - 1$.

> Indeed, there is nothing to prove if $k = 1$ (it is possible if and only if $g_0 = 0$), so that let us assume that $k > 1$ (i.e., that $g_0 \neq 0$). When $t < k - 1$, the dimensions of E_t and E_{t+1} clearly are t, $t + 1$, respectively, so that the inclusion $E_t \subset E_{t+1}$ is strict. Now, the vectors $g_0, Hg_0, ..., H^{k-2}g_0$ are linearly independent, and if we add to this family the vector $H^{k-1}g_0$, we get a linearly dependent set; it follows that the vector we add is a linear combination of the vectors $g_0, Hg_0, ..., H^{k-2}g_0$:
>
> $$H^{k-1}g_0 = \lambda_0 g_0 + \lambda_1 Hg_0 + ... + \lambda_{k-2}H^{k-2}g_0.$$
>
> Multiplying relation by H, H^2, H^3,..., we see that t-th Krylov vector, starting with $t = k$, is a linear combination of the previous Krylov vectors, whence (by induction) it is also a linear combination of the first $k - 1$ of these vectors. Thus, $E_t = E_{k-1}$ whenever $t \geq k$.

Now consider the affine sets

$$F_t = x_0 + E_t,$$

and let x_t be the minimizer of the quadratic form f on F_t. *By definition*, the trajectory of the Conjugate Gradient method as minimizing f, x_0 being the starting point, is the sequence $x_0, x_1, ..., x_{k-1}$. We are about to prove that

- x_{k-1} is the global minimizer of f;
- there exists explicit recurrence which allows to build sequentially the points $x_1, ..., x_{k-1}$.

7.3.1.2 Iterative representation of the Conjugate Gradient method

CG minimizes a strongly convex quadratic form

$$f(x) = \frac{1}{2}x^THx - b^Tx$$

as follows:

Algorithm 7.3.1 [Conjugate Gradient method]
Initialization: *choose arbitrary starting point x_0 and set*

$$d_0 = -g_0 \equiv -\nabla f(x_0) = b - Hx_0;$$

set $t = 1$.
Step t: *if* $g_{t-1} \equiv \nabla f(x_{t-1}) = 0$, *terminate,* x_{t-1} *being the result. Otherwise set*
[new iterate]

$$x_t = x_{t-1} + \gamma_t d_{t-1}, \quad \gamma_t = -\frac{g_{t-1}^T d_{t-1}}{d_{t-1}^T H d_{t-1}}, \tag{7.3.4}$$

[new gradient]

$$g_t = \nabla f(x_t) \equiv Hx_t - b, \tag{7.3.5}$$

[new direction]

$$d_t = -g_t + \beta_t d_{t-1}, \quad \beta_t = \frac{g_t^T H d_{t-1}}{d_{t-1}^T H d_{t-1}}, \tag{7.3.6}$$

replace t *with* $t+1$ *and loop.*

We are about to prove that the presented algorithm is a Conjugate Direction method:

Theorem 7.3.1 [Conjugate Gradient Theorem]
If the algorithm does not terminate at step t*, then*
(i_t) *The gradients* $g_0, ..., g_{t-1}$ *of* f *at the points* $x_0, ..., x_{t-1}$ *are nonzero and*

$$\mathrm{Lin}\{g_0, g_1, ..., g_{t-1}\} = E_t; \tag{7.3.7}$$

(ii_t) *The directions* $d_0, ..., d_{t-1}$ *are nonzero and*

$$\mathrm{Lin}\{d_0, ..., d_{t-1}\} = E_t; \tag{7.3.8}$$

(iii_t) *The directions* $d_0, ..., d_{t-1}$ *are* H*-orthogonal:*

$$d_i^T H d_j = 0, \ \ 0 \le i < j \le t-1; \tag{7.3.9}$$

(iv_t) x_t *is the minimizer of* f *on* F_t*, i.e.,* g_t *is orthogonal to* E_t
(v) *One has*

$$\gamma_t = \frac{g_{t-1}^T g_{t-1}}{d_{t-1}^T H d_{t-1}}; \tag{7.3.10}$$

(vi) *One has*

$$\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}}. \tag{7.3.11}$$

(vii) *The algorithm terminates no later than after* n *steps with the result being the exact minimizer of* f*.*

Proof.

1^0. We first prove (i)-(iv) by induction on t.

Base $t = 1$. We should prove that if the algorithm does not terminate at the first step (i.e., if $g_0 \neq 0$), then (i_1) - (iv_1) are valid. The only statement which should be proved is (iv_1), and to prove it is the same as to prove that $g_1 = \nabla f(x_1)$ is orthogonal to E_1. This is an immediate corollary of the following

Lemma 7.3.1 *Let* t *be such that* $g_{t-1} \neq 0$*. Then* $g_t^T d_{t-1} = 0$.

Proof of the Lemma. We have $\nabla f(x+u) = \nabla f(x) + Hu$, whence

$$g_t^T d_{t-1} = (g_{t-1} + \gamma_t H d_{t-1})^T d_{t-1} = 0$$

by definition of γ_t, see (7.3.4). □

Step $t \mapsto t+1$. Assume that (i_s) - (iv_s) are valid for $s \le t$ and that the algorithm does not terminate at the step $t+1$, i.e., that $g_t \neq 0$, and let us derive from this assumption (i_{t+1}) - (iv_{t+1}).

1^0. From (iv_s), $s \le t$, we know that g_s is orthogonal to E_s, and from (ii_s) and (i_s), the subspace E_s is the linear span of the vectors $d_0, ..., d_{s-1}$, same as it is the linear span of the vectors $g_0, ..., g_{s-1}$, and we conclude that g_s is orthogonal to the vectors $d_0, ..., d_{s-1}$:

$$g_s^T d_l = 0, \;\; 0 \le l < s \le t \tag{7.3.12}$$

and that the vectors $g_0, ..., g_t$ are mutually orthogonal:

$$g_s^T g_l = 0, \;\; 0 \le l < s \le t. \tag{7.3.13}$$

We have from (7.3.4)

$$g_t = Hx_t - b = H(x_{t-1} + \gamma_t d_{t-1}) - b = [Hx_{t-1} - b] + \gamma_t H d_{t-1} = g_{t-1} + \gamma_t H d_{t-1}.$$

By (i_{t-1}) and (ii_{t-1}), both g_{t-1} and d_{t-1} belong to $\mathrm{Lin}\{g_0, Hg_0, ..., H^{t-1}g_0\}$, so that the above computation demonstrates that $g_t \in \mathrm{Lin}\{g_0, Hg_0, ..., H^t g_0\}$, which combined with ($\mathrm{i}_t$) means that

$$\mathrm{Lin}\{g_0, ..., g_t\} \subset \mathrm{Lin}\{g_0, Hg_0, ..., H^t g_0\} = E_{t+1}.$$

Since $g_0, ..., g_t$ are nonzero and mutually orthogonal (see (7.3.13)), the left hand side subspace in the latter inclusion is of dimension $t+1$, while the right hand side subspace is of dimension at most $t+1$; a $t+1$-dimensional linear subspace can be enclosed into a linear subspace of dimension $\le t+1$ only if the subspaces are equal, and we come to (i_{t+1}).

To prove (ii_{t+1}), note that by (7.3.6)

$$d_t = -g_t + \beta_t d_{t-1};$$

both right hand side vectors are from $E_{t+1} = \mathrm{Lin}\{g_0, Hg_0, ..., H^t g_0\}$ (by (ii_t) and already proved (i_{t+1})), so that $d_t \in \mathrm{Lin}\{g_0, Hg_0, ..., H^t g_0\}$; combining this observation with (ii_t), we come to

$$\mathrm{Lin}\{d_0, ..., d_t\} \subset \mathrm{Lin}\{g_0, Hg_0, ..., H^t g_0\}. \tag{7.3.14}$$

Besides this, $d_t \neq 0$ (indeed, g_t is nonzero and is orthogonal to d_{t-1} by (7.3.12)). Now let us prove that d_t is H-orthogonal to $d_0, ..., d_{t-1}$. From formula for d_t we have

$$d_t^T H d_s = -g_t^T H d_s + \beta_t d_{t-1}^T H d_s. \tag{7.3.15}$$

When $s = t-1$, the right hand side is 0 by definition of β_t; and when $s < t-1$, both terms in the right hand side are zero. Indeed, the first term is zero due to (ii_t): this relation implies that

$$Hd_s \in \mathrm{Lin}\{Hg_0, H^2 g_0, ..., H^s g_0\},$$

and the right hand side subspace, due to (i_{s+1}) ($s < t-1$, so that we can use (i_{s+1})!), is contained in the linear span of the gradients $g_0, ..., g_{s+1}$, and g_t is orthogonal to all these gradients by virtue of (7.3.14) (recall that $s < t-1$). The second term in the right hand side of (7.3.15) vanishes because of (iii_t).

Thus, the right hand side in (7.3.15) is zero for all $s < t$; in other words, we have proved (iii_{t+1}) – the vectors $d_0, ..., d_t$ indeed are H-orthogonal. We already know that $d_t \neq 0$; consequently, in view of (ii_t), all $d_0, ..., d_t$ are nonzero. Since, as we already know, these vectors are H-orthogonal, they are linearly independent[4)]. Consequently, inclusion in (7.3.14) is in fact equality, and we have proved (ii_{t+1}).

It remains to prove (iv_{t+1}). We have $x_{t+1} = x_t + \gamma_{t+1} d_t$, whence

$$g_{t+1} = g_t + \gamma_{t+1} H d_t.$$

We should prove that g_{t+1} is orthogonal to E_{t+1}, i.e., due to already proved (iii_{t+1}), is orthogonal to every d_i, $i \leq t$. Orthogonality of g_{t+1} and d_t is given by Lemma 7.3.1, so that we should verify that $g_{t+1}^T d_i = 0$ for $i < t$. This is immediate:

$$g_{t+1}^T d_i = g_t^T d_i + \gamma_t d_t^T H d_i;$$

the first term in the right hand side is zero due to (7.3.12), and the second term is zero due to already proved (iii_{t+1}).

The inductive step is completed.

2^0. To prove (v), note that if $t > 1$ then, by (7.3.6),

$$-g_{t-1}^T d_{t-1} = g_{t-1}^T g_{t-1} - \beta_{t-1} g_{t-1}^T d_{t-2},$$

and the second term in the right hand side is 0 due to (7.3.13); thus, $-g_{t-1}^T d_{t-1} = g_{t-1}^T g_{t-1}$ for $t > 1$. For $t = 1$ this relation also is valid (since, by the initialization rule, $d_0 = -g_0$). Substituting equality $-g_{t-1}^T d_{t-1} = g_{t-1}^T g_{t-1}$ into formula for γ_t from (7.3.4), we get (7.3.10).

To prove (vi), note that $g_t^T g_{t-1} = 0$ (see (7.3.14). Besides this, from (7.3.4) we have

$$H d_{t-1} = \frac{1}{\gamma_t}[g_t - g_{t-1}]$$

(note that $\gamma_t > 0$ due to (v) and (i_t), so that we indeed can rewrite (7.3.4) in the desired way); taking inner product with g_t, we get

$$g_t^T H d_{t-1} = \frac{1}{\gamma_t} g_t^T g_t = \frac{d_{t-1}^T H d_{t-1}}{g_{t-1}^T g_{t-1}} g_t^T g_t$$

(we have used (7.3.10)); substituting the result into (7.3.6), we come to (7.3.11).

3^0. It remains to prove (vii). This is immediate: as we already know, if the method does not terminate at step t, i.e., if $g_{t-1} \neq 0$, then the vectors $g_0, ..., g_{t-1}$ are mutually orthogonal nonzero (and, consequently, linearly independent) vectors which form a basis in E_t; since there cannot be more than $k - 1 \leq n$ linearly independent vectors in E_t, k being the smallest t such that the Krylov vectors $g_0, Hg_0, ..., H^{t-1} g_0$ are linearly dependent, we see that the method terminates in no more than $k \leq n$ steps. And since it is terminated when $g_t = 0 = \nabla f(x_t) = 0$, the result of the method indeed is the global minimizer of f. ■

[4)] Indeed, we should verify that if

$$\sum_{i=0}^{t} \lambda_i d_i = 0,$$

then $\lambda_i = 0$, $i = 0, ..., t$. Multiplying the equality by H and then taking inner product of the result with d_i, we get

$$\lambda_i d_i^T H d_i = 0$$

(the terms $d_i^T H d_j$ with $i \neq j$ are zero due to H-orthogonality of $d_0, ..., d_t$), whence $\lambda_i = 0$ (since $d_i^T H d_i > 0$ due to $d_i \neq 0$ and to the positive definiteness of H). Thus, all coefficients λ_i indeed are zeros

7.3.1.3 CG and Three-Diagonal representation of a Symmetric matrix

Assume that we are minimizing (7.3.1) by the Conjugate Gradient algorithm; for the sake of simplicity, assume also that the method terminates in exactly n steps, so that in course of its run we obtain n nonzero gradients $g_0, ..., g_{n-1}$. As we know from the proof of the Conjugate Gradient Theorem (see (7.3.14)), these gradients are mutually orthogonal, so that after normalization

$$f_i = |g_i|^{-1} g_i$$

of g_i we get an orthonormal basis in $\mathbf{R}^n$. How does the matrix H look in this basis? The answer is very impressive:

the matrix in the basis $\{f_i\}$ is 3-diagonal: $f_i^T H f_j = 0$ whenever $|i-j| > 1$.

The proof is immediate: assume, e.g., that $i > j+1$ and let us prove that $f_i^T H f_j = 0$. As we know from Theorem 7.3.1.(i), $f_j \in \mathrm{Lin}\{g_0, Hg_0, ..., H^j g_0\}$, whence $Hf_j \in \mathrm{Lin}\{Hg_0, ..., H^{j+1}g_0\}$, and the latter subspace is contained in $\mathrm{Lin}\{g_0, ..., g_{j+1}\}$ (the same Theorem 7.3.1.(i)). Since, as it was already mentioned, g_i is orthogonal to $g_0, ..., g_{i-1}$ and $j+1 < i$, we have $f_i^T H f_j = 0$, as claimed.

The necessity to find an orthonormal basis where a given symmetric matrix becomes 3-diagonal occurs in many applications, e.g., when it is necessary to find the spectrum (all eigenvalues) of a large and sparse (with close to 0 percentage of nonzero entries) symmetric matrix. The Conjugate Gradient algorithm is certain "starting point" in developing tools for finding such a basis[5)].

7.3.1.4 Rate of convergence of the Conjugate Gradient method

According to Theorem 7.3.1, if CG as applied to (7.3.1) does not terminate during first t steps, x_t is the minimizer of f on the affine plane

$$F_t = x_0 + \mathrm{Lin}\{d_0, ..., d_{t-1}\} = x_0 + E_t,$$

where, according to (7.3.8),

$$E_t = \mathrm{Lin}\{g_0, Hg_0, ..., H^{t-1}g_0\}, \quad g_0 = Hx_0 - b. \tag{7.3.16}$$

Equivalent description of E_t is that this is the space comprised of all vectors of the form $p(H)g_0$, p being a polynomial of degree $\le t-1$:

$$E_t = \{p(H)g_0 \mid p(z) = \sum_{i=0}^{t-1} p_i z^i\}. \tag{7.3.17}$$

Given these observations, we immediately can establish the following

Proposition 7.3.1 *Let*

$$E(x) = f(x) - \min f$$

be the residual in terms of the objective associated with quadratic form (7.3.1), and let x_t be t-th iterate of the Conjugate Gradient method (if the method terminates in course of first t steps, then, by definition, x_t is the result of the method, i.e., the exact minimizer of f). Then

$$E(x_t) = \min_{p \in \mathcal{P}_{t-1}} \frac{1}{2}(x_0 - x^*)^T H[I - Hp(H)]^2 (x_0 - x^*), \tag{7.3.18}$$

where x_0 is the starting point, x^* is the exact minimizer of f, I is the unit matrix, and $\mathcal{P}_k$ is the family of all polynomials of degree $\le k$.

[5)] please do not think that the problem in question can be solved by straightforward application of the CG: the influence of rounding errors makes the actually computed gradients very far from being mutually orthogonal!

Proof is immediate: since f is a quadratic form, we have

$$f(x) = f(x^*) + (x-x^*)^T \nabla f(x^*) + \frac{1}{2}(x-x^*)^T [\nabla^2 f](x-x^*) = f(x^*) + \frac{1}{2}(x-x^*)^T H(x-x^*)$$

(we have taken into account that $\nabla f(x^*) = 0$ and $\nabla^2 f = H$), whence

$$E(x) = \frac{1}{2}(x-x^*)H(x-x^*). \tag{7.3.19}$$

Substituting

$$x = x_0 + p(H)g_0 = x_0 + p(H)(Hx_0 - b) = x_0 + p(H)H(x_0 - x^*),$$

we get $x - x^* = -[I - Hp(H)](x_0 - x^*)$, whence

$$E(x_0 + p(H)g_0) = \frac{1}{2}(x_0 - x^*)^T H[1 - Hp(H)]^2(x - x^*).$$

When p runs through the family $\mathcal{P}_{t-1}$, the point $x_0 + p(H)g_0$, as it already was explained, runs through the affine plane $F_t = x_0 + E_t$; x_t, as we know, is the minimizer of f (and, consequently, E) on this plane, and (7.3.18) follows. ∎

What we are interested in, is the following corollary of Proposition 7.3.1:

Corollary 7.3.1 *Let Σ be the spectrum (the set of distinct eigenvalues) of H, and let x_t be t-th point of the trajectory of CG as applied to f. Then*

$$E(x_t) \le E(x_0) \min_{q \in \mathcal{P}_t^*} \max_{\lambda \in \Sigma} q^2(\lambda), \tag{7.3.20}$$

where $\mathcal{P}_t^*$ is the set of all polynomials $q(z)$ of degree $\le t$ equal 1 at $z = 0$.

Besides this,

$$E(x_t) \le \left[\frac{1}{2}|x_0 - x^*|^2\right] \min_{q \in \mathcal{P}_t^*} \max_{\lambda \in \Sigma} \lambda q^2(\lambda). \tag{7.3.21}$$

Proof. Let $e_1, ..., e_n$ be an orthonormal basis comprised of eigenvectors of H, and let $\lambda_1, ..., \lambda_n$ be the corresponding eigenvalues. Let

$$x_0 - x^* = \sum_{i=1}^n s_i e_i$$

be the expansion of $x_0 - x^*$ in the indicated basis. Then, for any polynomial r,

$$r(H)(x_0 - x^*) = \sum_{i=1}^n r(\lambda_i) s_i e_i;$$

applying this identity to the polynomial $r(z) = z(1 - zp(z))^2$, $p \in \mathcal{P}_{t-1}$, we get

$$(x_0 - x^*)^2 H[1 - Hp(H)]^2(x_0 - x^*) = \left[\sum_{i=1}^n s_i e_i\right]^T \left[\sum_{i=1}^n \lambda_i (1 - \lambda_i p(\lambda_i))^2 s_i e_i\right] =$$

$$= \sum_{i=1}^n (1 - \lambda_i p(\lambda_i))^2 \lambda_i s_i^2. \tag{7.3.22}$$

The resulting quantity clearly can be bounded from above by

$$S \equiv \left[\max_{\lambda \in \Sigma} (1 - \lambda p(\lambda))^2\right] \sum_{i=1}^n \lambda_i s_i^2 = \left[\max_{\lambda \in \Sigma} (1 - \lambda p(\lambda))^2\right] \left[(x_0 - x^*)^T H(x_0 - x^*)\right] =$$

[see (7.3.19)].

$$= 2\left[\max_{\lambda\in\Sigma}(1-\lambda p(\lambda))^2\right] E(x_0).$$

Combining the resulting inequality with (7.3.18), we come to

$$E(x_t) \le E(x_0) \min_{p\in\mathcal{P}_{t-1}} \max_{\lambda\in\Sigma}(1-\lambda p(\lambda)^2.$$

When p runs through $\mathcal{P}_{t-1}$, the polynomial $q(z) = 1 - zp(z)$ clearly runs through the entire $\mathcal{P}_t^*$, and (7.3.20) follows.

Relation (7.3.21) is proved similarly; the only difference is that instead of bounding from above the right hand side of (7.3.22) by the quantity S, we bound the expression by

$$\left[\max_{\lambda\in\Sigma}\lambda(1-\lambda p(\lambda))^2\right]\sum_{i=1}^n s_i^2,$$

the second factor in the bound being $|x_0 - x^*|^2$. ■

Corollary 7.3.1 provides us with a lot of information on the rate of convergence of the Conjugate Gradient method:

- A. Rate of convergence in terms of Condition number It can be proved that for any segment $\Delta = [l, L]$, $0 < l < L < \infty$, and for any positive integer s there exists a polynomial $q_s \in \mathcal{P}_s^*$ with

$$\max_{\lambda\in\Delta} q_s^2(\lambda) \le 4\left[\frac{\sqrt{Q}-1}{\sqrt{Q}+1}\right]^{2s}, \quad Q = \frac{L}{l}.$$

  Combining (7.3.20) and the just indicated result where one should substitute

$$l = \lambda_{\min}(H), L = \lambda_{\max}(H),$$

  we get the following *non-asymptotical* efficiency estimate for CG as applied to (7.3.1):

$$E(x_N) \le 4\left[\frac{\sqrt{Q_H}-1}{\sqrt{Q_H}+1}\right]^{2N} E(x_0), \quad N = 1, 2, ... \tag{7.3.23}$$

  where

$$Q_H = \frac{\lambda_{\max}(H)}{\lambda_{\min}(H)}$$

  is the condition number of the matrix H.

  We came to a *non-asymptotical linear rate of convergence* with the convergence ratio

$$\left[\frac{\sqrt{Q_H}-1}{\sqrt{Q_H}+1}\right]^2;$$

  for large Q_h, this ratio is of the form $1 - O(1)\sqrt{Q_H}$, so that the number of steps required to improve initial inaccuracy by a given factor ϵ is, *independently of the value of* n, bounded from above by $O(1)\sqrt{Q_H}\ln(1/\epsilon)$. The number of required steps is proportional to the *square root* of the condition number of the Hessian, while for the Steepest Descent in the quadratic case similar quantity is proportional to the condition number itself (see Lecture 3); this indeed is a great difference!

- B. Rate of convergence in terms of $|x_0 - x^*|$ It can be proved also that for any $L > 0$ and any integer $s > 0$ there exists a polynomial $r_s \in \mathcal{P}_s^*$ such that

$$\max_{0 \le \lambda \le L} \lambda r^2(\lambda) \le \frac{L}{(2s+1)^2}.$$

Since Σ for sure is contained in the segment $[0, L = |H|]$, $|H|$ being the norm of matrix H, we can use just indicated result and (7.3.21) to get non-asymptotical (and independent on the condition number of H) sublinear efficiency estimate

$$E(x_N) \le \frac{|H||x_0 - x^*|^2}{2} \frac{1}{(2N+1)^2}, \quad N = 1, 2, ... \tag{7.3.24}$$

This result resembles sublinear global efficiency estimate for Gradient Descent as applied to a convex $C^{1,1}$ function (Lecture 9, Proposition 6.1.2); note that a convex quadratic form (7.3.1) indeed is a $C^{1,1}$ function with Lipschitz constant of gradient equal to $|H|$. As compared to the indicated result about Gradient Descent, where the convergence was with the rate $O(1/N)$, the convergence given by (7.3.24) is twice better in order – $O(1/N^2)$.

- C. Rate of convergence in terms of the spectrum of H The above results established rate of convergence of CG in terms of bounds – both lower and upper or only the upper one – on the eigenvalues of H. If we take into account details of the distribution of the eigenvalues, more detailed information on convergence rate can be obtained. Let

$$\lambda_{\max}(H) \equiv \lambda_1 > \lambda_2 > ... > \lambda_m \equiv \lambda_{\min}(H)$$

be *distinct* eigenvalues of H written down in descent order. For every $k \le s$ let

$$\pi_k(z) = \prod_{i=1}^{k} (1 - z/\lambda_i) \in \mathcal{P}_k^*,$$

and let $q_{s,k} \in \mathcal{P}_s^*$ be the polynomial such that

$$\max_{\lambda_{\min}(H) \le \lambda \le \lambda_{k+1}} q_{s,k}^2(\lambda) \le 4 \left[\frac{\sqrt{Q_{k,H}} - 1}{\sqrt{Q_{k,H}} + 1} \right]^{2s}, \quad Q_{k,H} = \frac{\lambda_{k+1}}{\lambda_{\min}(H)}$$

(existence of such a polynomial is mentioned in item A). It is clear that

$$\max_{\lambda \in \Sigma} [\pi_k(\lambda) q_{s,k}(\lambda)]^2 \le 4 \left[\frac{\sqrt{Q_{k,H}} - 1}{\sqrt{Q_{k,H}} + 1} \right]^{2s}$$

(indeed, π_k vanishes on the spectrum of H to the right of λ_{k+1} and is in absolute value ≤ 1 between zero and λ_{k+1}, while $q_{s,k}$ satisfies the required bound to the left of λ_{k+1} in Σ). Besides this, $\pi_k q_{s,k} \in \mathcal{P}_{k+s}^*$. Consequently, by (7.3.20)

$$E(x_N) \le 4 \min_{1 \le s \le N} \left\{ \left[\frac{\sqrt{Q_{N-s,H}} - 1}{\sqrt{Q_{N-s,H}} + 1} \right]^{2s} \right\} E(x_0)$$

– this is an extension of the estimate (7.3.23) (this latter estimate corresponds to the case when we eliminate the outer min and set $s = N$ in the inner brackets, which results in $Q_{0,H} = Q_H$). We see also that if $N \ge m$, m being the number of distinct eigenvalues in H, then $E(x_N) = 0$ (set $q = \pi_m$ in (7.3.20)); thus, in fact

CG finds the exact minimizer of f in at most as many steps as many distinct eigenvalues are there in matrix H.

Taking into account the details of the spectrum of H, one can strengthen the estimate of item B as well.

7.3.1.5 Conjugate Gradient algorithm for quadratic minimization: advantages and disadvantages

Let us summarize our knowledge on the CG algorithm for (strongly convex) quadratic minimization, or, which is the same, for solving linear systems

$$Hx = b$$

with positive definite symmetric matrix H.

First of all, note that the method is simple in implementation – as simple as the Gradient Descent: a step of the method requires not more than 2 multiplications of vectors by matrix A. Indeed, literally reproducing formulae (7.3.4) - (7.3.6), you need 2 matrix-vector multiplications:

$d_{t-1} \to Hd_{t-1}$ to find γ_t,

and

$x_t \to Hx_t$ to find g_t.

In fact only the first of these matrix-vector multiplication is necessary, since g_t can be computed recursively:

$$g_t = g_{t-1} + \gamma_t H d_{t-1} \quad [\text{since } \; g_t - g_{t-1} = H(x_t - x_{t-1}) = \gamma_t H d_{t-1}].$$

Note that all remaining actions at a step are simple – taking inner products and linear combinations of vectors, and all these actions together cost $O(n)$ arithmetic operations. Thus, the arithmetic cost of a step in CG (same as that one for Steepest Descent) is

$$O(n) + [\text{cost of a single matrix-vector multiplication } d \to Hd].$$

This is a very important fact. It demonstrates that *sparsity of H* – relatively small number N ($N << n^2$) of nonzero entries – can be immediately utilized by CG. Indeed, in the "dense" case matrix-vector multiplication costs $2n^2$ multiplications and additions, and this is the principal term in the arithmetic cost of a step of the method; in the sparse case this principal term reduces to $2N << 2n^2$.

Large-scale linear systems of equations typically have matrices H which either are extremely sparse (something $0.01\% - 1\%$ of nonzero entries), or are not sparse themselves, but are products of two sparse matrices ("implicit sparsity", e.g., the least square matrices arising in Tomography); in this latter case matrix-vector multiplications are as cheap as if H itself were sparse. If the size of the matrix is large enough (tens of thousands; in Tomography people deal with sizes of order of 10^5 - 10^6) and no sparsity – explicit or "implicit" – is present, then, typically, there are no ways to solve the system. Now, if the matrix of the system is sparse and the pattern of the nonzero entries is good enough, one can solve the system by a kind of Choleski decomposition or Gauss elimination, both the methods being modified in order to work with sparse data and not to destroy sparsity in course of their work. If the matrix is large and sparsity is not "well-structured" or is "implicit", the direct methods of Linear Algebra are unable to solve the system, and all we can do is to use an iterative method, like Steepest Descent or CG. Here we exploit the main advantage of an iterative method based on matrix-vector multiplications – cheap step and modest memory requirements.

The indicated advantages of iterative methods are shared by both Steepest Descent and Conjugate Gradient. But there is an important argument in favour of CG – its better rate of convergence. In fact, *the Conjugate Gradient algorithm possesses the best*, in certain exact sense, *rate of convergence an iterative method* (i.e., the one based on matrix-vector multiplications) *may have*.

These are the main advantages of the CG – simplicity and theoretical optimality among the iterative methods for quadratic minimization. And the main disadvantage of the method is

its sensitivity to the condition number of the matrix of the system – although less than the one for Steepest Descent (see item A in the discussion above), but still rather unpleasant. Theoretically, all bad we could expect of an ill-conditioned H is that the convergence of CG will be slow, but after n steps (as always, n is the size of H) the method should magically bring us the exact solution. The influence of rounding errors makes this attractive picture absolutely unrealistic. Even with moderate condition number of H, the method will not find exact solution in n steps, but will come rather close to it; and with large condition number, the n-th approximate solution can be even worse than the initial one. Therefore when people, by some reasons, are interested to solve a moderate-size linear system by CG, they allow the method to run $2n$, $4n$ or something like steps (I am saying about "moderate size" systems, since for large-scale ones $2n$ or $4n$ steps of CG simply cannot be carried out in reasonable time).

The conclusion here should be as follows: if you are solving a linear system with symmetric positive definite matrix and the size of the system is such that direct Linear Algebra methods – like Choleski decomposition – can be run in reasonable time, it is better to use these direct methods, since they are much more numerically stable and less sensitive to the conditioning of the system than the iterative methods. It makes sense to solve the system by CG only when the direct methods cannot be used, and in this case your chances to solve the problem heavily depend on whether you can exploit explicit or implicit sparsity of the matrix in question and especially on how well-conditioned is the matrix.

7.3.2 Extensions to non-quadratic problems

To the moment we in fact dealt not with unconstrained minimization, but with the Linear Algebra problem of solving a linear system with positive definite symmetric matrix. All this work in fact was aimed to develop extensions of the Conjugate Gradient algorithm on the nonquadratic case, and it is time now to come to these extensions.

The idea behind the extensions in question is as follows: the Conjugate Gradient Algorithm in its basic version 7.3.1:

(A): $g_0 = \nabla f(x_0);\ d_0 = -g_0;$

(B): $x_t = x_{t-1} + \gamma_t d_{t-1}, \quad \gamma_t = -\frac{g_{t-1}^t d_{t-1}}{d_{t-1}^T H d_{t-1}};$

(C): $g_t = \nabla f(x_t)$:

(D): $d_t = -g_t + \beta_t d_{t-1}, \quad \beta_t = \frac{g_t^T H d_{t-1}}{d_{t-1}^T H d_{t-1}};$

"almost ignores" the quadratic nature of f: the matrix H is involved only in the formulae for scalars γ_t (the stepsize) and β_t (the coefficient in the updating formulae for the search directions). If we were able to eliminate the presence of H completely and to describe the process in terms of f and ∇f only, we would get a recurrence CG* which, formally, could be applied to an arbitrary objective f, and in the case of strongly convex quadratic objective would become our basic method CG. This latter method solves quadratic problem *exactly* in n steps; since close to a nondegenerate local minimizer x^* a general smooth objective f is very similar to a strongly convex quadratic one f_q, we could hope that CG* applied to f and started close to x^* would "significantly" reduce inaccuracy in n steps. Now we could again apply to f n steps of the same routine CG*, but with the starting point given by the first n steps, again hopefully significantly reducing inaccuracy, and so on. If, besides this, we were clever enough to ensure global convergence of the indicated "cyclic" routine, we would get a globally converging method with good asymptotical behaviour.

This is the idea, and now let us look how to implement it. First of all, we should eliminate H in formulae for the method in the quadratic case. It is easy to do it with the formula for the stepsize γ_t. Indeed, we know from Lemma 7.3.1 that in the strongly convex quadratic case g_t is orthogonal to d_{t-1}, so that x_t is the minimizer of f along the line passing through x_{t-1} in the direction d_{t-1}. Thus, we may replace (B) by equivalent, in the case of strongly convex quadratic f, rule

(B*): $x_t = x_{t-1} + \gamma_t d_{t-1}, \quad \gamma_t \in \mathrm{Argmin}_{\gamma \in \mathbf{R}} f(x_{t-1} + \gamma d_{t-1});$

this new rule makes sense for a non-quadratic f as well.

It remains to eliminate H in (D). This can be done, e.g., via the identity given by Theorem 7.3.1.(vi):

$$\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}}.$$

With these substitutions, we come to the following

Algorithm 7.3.2 [Fletcher-Reeves Conjugate Gradient method for minimization of a general-type function f over $\mathbf{R}^n$]
Initialization: *choose arbitrary starting point x_0. Set cycle counter $k = 1$.*
Cycle k:

Initialization of the cycle: *given x_0, compute*

$$g_0 = \nabla f(x_0), \quad d_0 = -g_0;$$

Inter-cycle loop: *for $t = 1, ..., n$:*

a) if $g_{t-1} = 0$, terminate, x_{t-1} being the result produced by the method, otherwise set $x_t = x_{t-1} + \gamma_t d_{t-1}$, where γ_t minimizes $f(x_{t-1} + \gamma d_{t-1})$ over $\gamma \in \mathbf{R}$;

b) compute $g_t = \nabla f(x_t)$;

c) set $d_t = -g_t + \beta_t d_{t-1}$, with $\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}}$.

If $t < n$, replace t with $t + 1$ and go to a)

Restart: *replace x_0 with x_n, replace k with $k + 1$ and go to new cycle.*

The Fletcher-Reeves algorithm is not the only extension of the quadratic Conjugate Gradient algorithm onto non-quadratic case. There are many other ways to eliminate H from Algorithm 7.3.1, and each of them gives rise to a non-quadratic version of CG. For example, one can rewrite the relation

$$\beta_t = \frac{g_t^T g_t}{g_{t-1}^T g_{t-1}} \tag{7.3.25}$$

equivalently in the quadratic case as

$$\beta_t = \frac{(g_t - g_{t-1})^T g_t}{g_{t-1}^T g_{t-1}} \tag{7.3.26}$$

(as we remember from the proof of Theorem 7.3.1, in the quadratic case g_t is orthogonal to g_{t-1}, so that both equations for β_t are in this case equivalent). When we replace the formula for β_t in the Fletcher-Reeves method by (7.3.26), we again obtain a method (the *Polak-Ribiere* one) for unconstrained minimization of smooth general-type functions, and this method also becomes the quadratic CG in the case of quadratic objective. It should be stressed that in the nonquadratic case the Polak-Ribiere method differs from the Fletcher-Reeves one, since relations (7.3.25) and (7.3.26) are equivalent only in the case of quadratic f.

7.3.3 Global and local convergence of Conjugate Gradient methods in non-quadratic case

Proposition 7.3.2 *[Global convergence of the Fletcher-Reeves and the Polak-Ribiere methods] Let a continuously differentiable function $f : \mathbf{R}^n \to \mathbf{R}$ be minimized by the Fletcher-Reeves or the Polak-Ribiere versions of the Conjugate Gradient method. Assume that the starting point x_0 of the first cycle is such that the level set*

$$S = \{x \mid f(x) \leq f(x_0)\}$$

is bounded, and let x^k be the starting point of cycle k (i.e., the result of the previous cycle). Then the sequence $\{x^k\}$ is bounded, and all limiting points of this sequence are critical points of f.

The proof, basically, repeats the one for the Steepest Descent, and I shall only sketch it. The first observation is that the objective never increases along the trajectory, since all steps of the method are based on precise line search. In particular, the trajectory never leaves the compact set S.

Now, the crucial observation is that the first step of cycle k is the usual Steepest Descent step from x^k and therefore it "significantly" decreases f, provided that the gradient of f at x^k is not very small [6)]. Since the subsequent inter-cycle steps do not increase the objective, we conclude that the sum of progresses in the objective values at the Steepest Descent steps starting the cycles is bounded from above (by the initial residual in terms of the objective). Consequently, these progresses tend to zero as $k \to \infty$, and due to the aforementioned relation

$$\text{small progress in the objective at a Steepest Descent step from } x^{k-1} \Rightarrow$$

$$\Rightarrow \text{small } |\nabla f(x^{k-1})|$$

we conclude that $\nabla f(x^k) \to 0$, $k \to \infty$. Thus, any limiting point of the sequence $\{x^k\}$ is a critical point of f.

The actual justification of non-quadratic extensions of the Conjugate Gradient method is the following proposition which says that the property "finite n-step convergence" of the method in the quadratic case transforms into the property of "n-step quadratic convergence" in the nonquadratic one:

Proposition 7.3.3 [Asymptotical "n-step quadratic" convergence of the Fletcher-Reeves and Polak-Ribiere methods in nonquadratic nondegenerate case]
Let $f : \mathbf{R}^n \to \mathbf{R}$ be three times continuously differentiable function, and let x^* be a nondegenerate local minimizer of f, so that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Assume that f is minimized by the Fletcher-Reeves or Polak-Ribiere versions of the Conjugate Gradient algorithm, and assume that the sequence $\{x^k\}$ of points starting the cycles of the algorithm converges to x^*. Then the sequence $\{x^k\}$ converges to x^* quadratically:

$$|x^{k+1} - x^*| \le C|x^k - x^*|^2$$

for certain $C < \infty$ and all k.

We should stress that the quadratic convergence indicated in the theorem is *not* the quadratic convergence of the subsequent search points generated by the method: in the Proposition we speak about "squaring the distance to x^*" in n steps of the method, not after each step, and for large n this "n-step quadratic convergence" is not that attractive.

7.4 Quasi-Newton Methods

The quasi-Newton methods, same as the Conjugate Gradient ones, are aimed to imitate the behaviour of the Newton method, avoiding at the same time usage of the second order derivatives and solving Newton systems of linear equations.

[6)] all is going on within a compact set S where f is continuously differentiable; therefore for $x \in S$ we have

$$f(x+h) \le f(x) + h^T \nabla f(x) + \epsilon(|h|)|h|, \ \ \forall (h, |h| \le 1),$$

with *independent of* $x \in S$ reminder $\epsilon(s) \to 0$, $s \to 0$. Consequently, properly chosen step from a point $x \in S$ in the anti-gradient direction indeed decreases f at least by $\psi(|\nabla f(x)|)$, with some positive on the ray $s > 0$ and nondecreasing on the ray function $\psi(s)$

7.4.1 The idea

The quasi-Newton methods belong to the generic family of *variable metric routines* known to us from Section 7.2.1. Recall that the generic Variable Metric Algorithm 7.2.1 is the recurrence of the type

$$x_{t+1} = x_t - \gamma_{t+1} S_{t+1} \nabla f(x_t), \tag{7.4.1}$$

where S_{t+1} are symmetric positive definite matrices (in the notation of Algorithm 7.2.1, $S_{t+1} = A_{t+1}^{-1}$). As about stepsizes γ_{t+1}, they are given by a kind of line search in the direction

$$d_{t+1} = -S_{t+1} \nabla f(x_t);$$

these directions are descent for f at non-critical points x_t:

$$\nabla f(x_t) \neq 0 \Rightarrow d_{t+1}^T \nabla f(x_t) \equiv -(\nabla f(x_t))^T S_{t+1} \nabla f(x_t) < 0 \tag{7.4.2}$$

(S_t is positive definite!).

As we remember from Section 7.2.1, a good choice of S_t should make the matrices $A_{t+1} = S_{t+1}^{-1}$ "positive definite corrections" of the Hessians $\nabla^2 f(x_t)$, and the Modified Newton methods (Section 7.2) more or less straightforwardly implemented this idea: there we sequentially

- computed the Hessians $\nabla^2 f(x_t)$,
- modified them, if necessary, to make the resulting matrices A_{t+1} "well positive definite",

  and, finally,
- computed $S_{t+1} = A_{t+1}^{-1}$ to get the search directions d_{t+1}.

In the quasi-Newton methods we use another approach to implement the same idea: we compute the matrices S_{t+1} recursively without explicit usage of the Hessians of the objective and inverting these Hessians. The recurrence defining S_{t+1} is aimed to ensure, at least in good cases, that

$$S_{t+1} - [\nabla^2 f(x_t)]^{-1} \to 0, \ t \to \infty, \tag{7.4.3}$$

which, basically, means that the method asymptotically becomes close to the Newton one and therefore quickly converges.

The problem is how to ensure (7.4.3) at a "cheap" computational cost.

7.4.2 The Generic Quasi-Newton Scheme

The idea behind the below policies for ensuring (7.4.3) is very simple. We intend to generate S_t recursively. Namely, assume that at the step t (when updating x_{t-1} into x_t) we used certain approximation S_t of the matrix

$$[\nabla^2 f(x_{t-1})]^{-1},$$

so that x_t was obtained from x_{t-1} by certain step in the direction

$$-S_t g_{t-1}, \quad g_s \equiv \nabla f(x_s).$$

Thus,

$$p_t \equiv x_t - x_{t-1} \equiv -\gamma_t S_t g_{t-1}, \tag{7.4.4}$$

γ_t being the stepsize given by linesearch. The first-order information on f at the points x_t and x_{t-1} allows us to define the vectors

$$q_t \equiv g_t - g_{t-1}. \tag{7.4.5}$$

If the step p_t is small in norm (as it should be the case at the final stage) and f is twice continuously differentiable (which we assume from now on), then

$$q_t \equiv g_t - g_{t-1} \equiv \nabla f(x_t) - \nabla f(x_{t-1}) \approx [\nabla^2 f(x_{t-1})](x_t - x_{t-1}) \equiv [\nabla^2 f(x_{t-1})] p_t; \tag{7.4.6}$$

of course, if f is quadratic, $\approx$ is simply an equality, independently of whether p_t is small or not.

Thus, after the step t we have enriched our information on the Hessian: now we have not only the previous approximation S_t to $[\nabla^2 f(x_{t-1})]^{-1}$, but also some approximation (namely, q_t) to the vector $[\nabla^2 f(x_{t-1})]p_t$. We can use this additional information to update the previous approximation to the inverse Hessian. The simplest and the most natural idea is to impose on S_{t+1} the following condition (which is an equality version of the approximate equality (7.4.6)):

$$S_{t+1}q_t = p_t.$$

This relation, of course, can be satisfied by infinitely many updatings $S_t \mapsto S_{t+1}$, and there are different "natural" ways to specify this updating. When the policy of the updatings is fixed, we get a particular version of the generic Variable Metric algorithm, up to the freedom in the linesearch tactics. We shall always assume in what follows that this latter issue – which linesearch to use – is resolved in favour of the exact linesearch. Thus, the generic algorithm we are going to consider is as follows:

Algorithm 7.4.1 [Generic Quasi-Newton method]
Initialization: *choose somehow starting point x_0, the initial positive definite symmetric matrix S_1, compute $g_0 = \nabla f(x_0)$ and set $t = 0$.*
Step t: *given x_{t-1}, $g_{t-1} = \nabla f(x_{t-1}) \neq 0$ and S_t, check whether $g_{t-1} = 0$. If it is the case, terminate (x_{t-1} is a critical point of f), otherwise*

- *set*
$$d_t = -S_t g_{t-1};$$
- *perform exact line search from x_{t-1} in the direction d_t, thus getting new iterate*
$$x_t = x_{t-1} + \gamma_t d_t;$$
- *compute $g_t = \nabla f(x_t)$ and set*
$$p_t = x_t - x_{t-1},\ q_t = g_t - g_{t-1};$$
- (U): *update S_t into positive definite symmetric matrix S_{t+1}, maintaining the relation*
$$S_{t+1}q_t = p_t, \tag{7.4.7}$$
replace t with $t+1$ and loop.

When specifying the only "degree of freedom" in the presented generic algorithm, i.e., the rules for (U), our targets are at least the following:

- (A) the matrices S_t should be symmetric positive definite;
- (B) in the case of strongly convex quadratic f the matrices S_t should converge (ideally – in finite number of steps) to the inverse Hessian $[\nabla^2 f]^{-1}$.

The first requirement is the standard requirement for Variable Metric algorithms; it was motivated in Section 7.2.1. The second requirement comes from our final goal – to ensure, at least in good cases (when the trajectory converges to a nondegenerate local minimizer of f), that $S_{t+1} - [\nabla^2 f(x_t)]^{-1} \to 0$, $t \to \infty$; as we remember, this property underlies fast – superlinear – asymptotical convergence of the algorithm.

Implementing (U) in a way which ensures the indicated properties and incorporating, in addition, certain policies which make the algorithm globally converging (e.g., restarts, which we already have used in non-quadratic extensions of the Conjugate Gradient method), we will come to globally converging methods with nice, in good cases, asymptotical convergence properties.

In the rest of the Lecture we focus on two frequently used forms of updating (U)

7.4.3 Implementations

7.4.3.1 Davidon-Fletcher-Powell method

In this method, (U) is given by

$$S_{t+1} = S_t + \frac{1}{p_t^T q_t} p_t p_t^T - \frac{1}{q_t^T S_t q_t} S_t q_t q_t^T S_t. \tag{7.4.8}$$

We are about to demonstrate that (7.4.8) is well-defined, results in positive definite S_{t+1} and maintains (7.4.7).

Proposition 7.4.1 *Let R be a positive definite symmetric matrix, and let p and q be two vectors such that*

$$p^T q > 0. \tag{7.4.9}$$

Then the matrix

$$R' = R + \frac{1}{p^T q} p p^T - \frac{1}{q^T R q} R q q^T R \tag{7.4.10}$$

is symmetric positive definite and satisfies the relation

$$R' q = p. \tag{7.4.11}$$

Proof. 1^0. Let us prove that R' satisfies (7.4.11). Indeed, we have

$$R'q = Rq + \frac{1}{p^T q}(p^T q)p - \frac{1}{q^T R q}(q^T R q)Rq = Rq + p - Rq = p.$$

2^0. It remains to verify that R' is positive definite. Let x be an arbitrary nonzero vector; we should prove that $x^T R' x > 0$. Indeed,

$$x^T R' x = x^T R x + \frac{(x^T p)^2}{p^T q} - \frac{(x^T R q)^2}{q^T R q};$$

setting $a = R^{1/2}x, b = R^{1/2}q$, we can rewrite the relation as

$$x^T R' x = \frac{(a^T a)(b^T b) - (a^T b)^2}{b^T b} + \frac{(x^T p)^2}{p^T q}.$$

Since, by assumption, $p^T q > 0$, the second fraction is nonnegative. The first one also is nonnegative by Cauchy's inequality. Thus, $x^T R' x \geq 0$, while we need to prove that this quantity is positive. To this end it suffices to verify that both numerators in the right hand side of the latter relation cannot vanish simultaneously. Indeed, the first numerator can vanish only if a is proportional to b (this is said by Cauchy's inequality: it becomes equality only when the vectors in question are proportional). If a is proportional to b, then x is proportional to q (see the origin of a and b). But if $x = sq$ for some nonzero (x is nonzero!) s, then the second numerator is $s^2(q^T p)^2$, and we know that $q^T p$ is positive. ∎

Now we can prove that (7.4.8) indeed can be used as (U):

Proposition 7.4.2 *Let at a step t of Algorithm 7.4.1 with (U) given by (7.4.8) the matrix S_t be positive definite and $g_{t-1} = \nabla f(x_{t-1}) \neq 0$ (so that x_{t-1} is not a critical point of f, and the step t indeed should be performed). Then S_{t+1} is well-defined, is positive definite and satisfies (7.4.7).*

Proof. It suffices to verify that $q_t^T p_t > 0$; then we would be able to get all we need from Proposition 7.4.1 (applied with $R = S_t, q = q_t, p = p_t$).

Since $g_{t-1} \neq 0$ and S_t is positive definite, the direction $d_t = -S_t g_{t-1}$ is a descent direction of f at x_{t-1} (see (7.4.2)). Consequently, the exact linesearch results in a nonzero stepsize, and $p_t = x_t - x_{t-1} = \gamma_t d_t \neq 0$. We have

$$q_t^T p_t = \gamma_t (g_t - g_{t-1})^T d_t =$$

[since x_t is a minimizer of f on the ray $\{x_{t-1} + \gamma d_t \mid \gamma > 0\}$ and therefore g_t is orthogonal to the direction d_t of the ray]

$$= -\gamma_t g_{t-1}^T d_t = \gamma_t g_{t-1}^T S_t g_{t-1} > 0$$

(S_t is positive definite). ∎

It can be proved that the Davidon-Fletcher-Powell method, as applied to a strongly convex quadratic form, finds exact solution in no more than n steps. n being the dimension of the design vector. Moreover the trajectory generated by the method initialized with $S_1 = I$ is exactly the one of the Conjugate Gradient method, so that the DFP (Davidon-Fletcher-Powell) method with the indicated initialization is a Conjugate Gradient method – in the quadratic case it becomes the standard Conjugate Gradient.

7.4.3.2 The Broyden family

The DFP version of (U) is certain rank 2 formula – the updated matrix S_{t+1} differs from S_t by a matrix of rank 2. The *Broyden* family of quasi-Newton methods is based on another rank 2 updating formula – the *Broyden-Fletcher-Goldfarb-Shanno* one.

The formula is as follows (those interested in its derivation are referred to Appendix to the lecture):

$$\text{(I)} \quad S_{t+1}^{BFGS} = S_t + \frac{1 + q_t^T S_t q_t}{(p_t^T q_t)^2} p_t p_t^T - \frac{1}{p_t^T q_t} \left[p_t q_t^T S_t + S_t q_t p_t^T \right]$$

(we write ^{BFGS} to indicate that this is the Broyden-Fletcher-Goldfarb-Shanno updating). It can be proved (see Proposition 7.4.3 in Appendix) that (I) is a correct version of (U): independently of what is positive definite symmetric S_t, (I) results in positive definite symmetric S_{t+1} satisfying the relation

$$(*) \qquad S_{t+1} q_t = p_t,$$

exactly as the Davidon-Fletcher-Powell updating

$$\text{(II)} \quad S_{t+1}^{DFP} = S_t + \frac{1}{q_t^T p_t} p_t p_t^T - \frac{1}{q_t^T S_t q_t} S_t q_t q_t^T S_t.$$

Now we have two updating formulae – (I) and (II); they transform a positive definite matrix S_t into positive definite matrices S_{t+1}^{BFGS} and S_{t+1}^{DFP}, respectively, satisfying (*). Since (*) is linear in S_{t+1}, *any convex combination of the resulting matrices also satisfies (*).* Thus, we come to the *Broyden* implementation of (U) given by

$$S_{t+1}^{\phi} = (1 - \phi) S_{t+1}^{DFP} + \phi S_{t+1}^{BFGS}; \tag{7.4.12}$$

here $\phi \in [0, 1]$ is the parameter specifying the updating. Note that the particular case of $\phi = 0$ corresponds to the Davidon-Fletcher-Powell method.

One can verify by a direct computation that

$$S_{t+1}^{\phi} = S_{t+1}^{DFP} + \phi v_{t+1} v_{t+1}^T, \quad v_{t+1} = (q_t^T S_t q_t)^{1/2} \left[\frac{1}{p_t^T q_t} p_t - \frac{1}{q_t^T S_t q_t} S_t q_t \right]. \tag{7.4.13}$$

From the considerations which led us to (7.4.12), we get the following

Corollary 7.4.1 *A Broyden method, i.e., Algorithm 7.4.1 with (U) given by (7.4.12), $\phi \in [0, 1]$ being the parameter of the method (which may vary from iteration to iteration), is a quasi-Newton method: it maintains symmetry and positive definiteness of the matrices S_t and ensures (7.4.7).*

It can be proved that, as applied to a strongly convex quadratic form f, the Broyden method minimizes the form exactly in no more than n steps, n being the dimension of the design vector, and if S_0 is proportional to the unit matrix, then the trajectory of the method on f is exactly the one of the Conjugate Gradient method.

There is the following remarkable fact:

all Broyden methods, independently of the choice of the parameter ϕ, being started from the same pair (x_0, S_1), equipped with the same exact line search and being applied to the same problem, generate the same sequence of iterates (although not the same sequence of matrices H_t!).

Thus, in the case of exact line search all methods from the Broyden family are the same. The methods became different only when inexact line search is used; although inexact line search is forbidden by our theoretical considerations, it is always the case in actual computations.

Broyden methods are thought to be the most efficient practically versions of the Conjugate Gradient and quasi-Newton methods. After intensive numerical testing of different policies of tuning the parameter ϕ, it was found that the best is the simplest policy $\phi \equiv 1$, i.e., the pure Broyden-Fletcher-Goldfarb-Shanno method.

Remark 7.4.1 Practical versions of the Broyden methods.

In practical versions of Broyden methods, exact line search is replaced with inexact one. Besides the standard requirements of "significant decrease" of the objective in course of the line search (like those given by the Armijo test), here we meet with specific additional requirement: the line search should ensure the relation

$$p_t^T q_t > 0. \tag{7.4.14}$$

In view of Proposition 7.4.1 this relation ensures that the updating formulae (I) and (II) (and, consequently, the final formula (7.4.12) with $\phi \in [0,1]$) maintain positive definiteness of S_t's and relation (7.4.7), i.e., the properties crucial for the quasi-Newton methods.

Relation (7.4.14) is ensured by the exact line search (we know this from the proof of Proposition 7.4.2), but of course not only by it: the property is given by a strict inequality and therefore, being valid for the stepsize given by the exact line search, is for sure valid if the stepsize is close enough to the "exact" one.

Another important implementation issue is as follows. Under assumption (7.4.14), updating (7.4.12) should maintain positive definiteness of the matrices S_t. In actual computations, anyhow, rounding errors may eventually destroy this crucial property, and when it happens, the method may become behave itself crazy. To overcome this difficulty, in good implementations people store and update from step to step not the matrices S_t themselves, but their Choleski factors: lower-triangular matrices C_t such that $S_t = C_t C_t^T$, or, more frequently, the Choleski factors C'_t of the inverse matrices S_t^{-1}. Updating formula (7.4.12) implies certain routines for updatings $C_t \mapsto C_{t+1}$ (respectively, $C'_t \mapsto C'_{t+1}$), and these are the formulae in fact used in the algorithms. The arithmetic cost of implementing such a routine is $O(n^2)$, i.e., is of the same order of complexity as the original formula (7.4.12); on the other hand, it turns out that this scheme is much more stable with respect to rounding errors, as far as the descent properties of the actually computed search directions are concerned.

7.4.4 Convergence of Quasi-Newton methods

7.4.4.1 Global convergence

In practice, quasi-Newton methods are usually executed in *continuous fashion:* Algorithm 7.4.1 is started at certain x_0 with certain positive definite S_1 and is run "forever" with the chosen version of (U). For such a scheme, *global convergence* is proved only for certain versions of the method and only under strong assumptions on f.

Of course, there is no difficulty in proving global convergence for the *scheme with restarts*, where one resets current S_t after every m steps to the initial (positive definite) value of the matrix; here m is certain fixed "cycle duration" (compare with the non-quadratic Conjugate Gradient methods from the previous Section). For the latter scheme, it is easy to prove that if the level set

$$L = \{x \mid f(x) \leq f(x_0)\}$$

associated with the initial point is bounded, then the trajectory is bounded and all limiting points of the sequence $\{x_{mk}\}_{k=0}^{\infty}$ are critical points of f. The proof is quite similar to the one of Proposition 7.3.2: the steps with indices $1 + mt$ are

$$x_{mk+1} \in \mathrm{Argmin}\{f(x_{mk} - \gamma S \nabla f(x_{mk})) \mid \gamma \geq 0\},$$

S being a once for ever fixed symmetric positive definite matrix (the one to which S_t is reset at the restart steps). Such a step decreases the objective "significantly", provided that $\nabla f(x_{mk})$ is not small[7)]; this property can be immediately derived (try to do it!) from positive definiteness of S, continuous differentiability of the objective and boundedness of L. At the remaining steps the objective never increases due to the linesearch, and we conclude that the sum over t of progresses $f(x_{t-1}) - f(x_t)$ in the objective value is bounded. Thus, the progress at step t tends to 0 as $t \to \infty$, and, consequently, $\nabla f(x_{mk}) \to 0$ as $k \to \infty$ – otherwise, as it was indicated, the progresses at the restart steps could not tend to 0 as $k \to \infty$. Thus, $\nabla f(x_{mk}) \to 0$, $k \to \infty$, so that every limiting point of the sequence $\{x_{mk}\}_k$ indeed is a critical point of f.

7.4.4.2 Local convergence

As far as *local* convergence is concerned, we, as always, are interested in the following question:

> let x^* be a nondegenerate local minimizer of f (i.e., $\nabla f(x^*) = 0$, $\nabla^2 f(x^*)$ is positive definite). Assume that the trajectory of the method in question converges to x^*; what can be said about the asymptotical rate of convergence?

We are about to answer this question for the most frequently used BFGS method (recall that it is the Broyden method with $\phi \equiv 1$).

First of all, consider the scheme with restarts and assume, for the sake of simplicity, that $m = n$ and $S = I$, S being the matrix to which S_t is reset after every n steps. Besides this, assume that the objective is smooth – three times continuously differentiable. In this case one can expect that the convergence of the sequence $\{x_{mk}\}_{k=0}^{\infty}$ to x^* will be quadratic. Indeed, in the strongly convex quadratic case the BFGS method with the initialization in question becomes the Conjugate Gradient method, so that in fact we are speaking about a nonquadratic extension of the CG and can use the reasoning from the previous section. Our expectations indeed turn out to be valid.

Anyhow, the "squaring the distance to x^* in every n steps" is not that attractive property, especially when n is large. Moreover, the scheme with restarts is not too reasonable from the asymptotical point of view: all our motivation of the quasi-Newton scheme came from the desire to ensure in a good case (when the trajectory converges to a nondegenerate local minimizer of f) the relation $S_t - \nabla^2 f(x_t) \to 0$, $t \to \infty$, in order to make the method asymptotically similar to the Newton one; and in the scheme with restarts we destroy this property when resetting S_t to the unit matrix at the restarting steps. To utilize the actual potential of the quasi-Newton scheme, we should avoid restarts, at least asymptotically (at the beginning they might become necessary to ensure global convergence).

It is very difficult to prove something good about local convergence of a quasi-Newton method executed in "continuous fashion" without restarts. There is, anyhow, the following remarkable result of Powell (1976):

[7)] here is the exact formulation: for every $\epsilon > 0$ there exists $\delta > 0$ such that if, for some k, $|\nabla f(x_{mk})| \geq \epsilon$, then $f(x_{mk+1}) \leq f(x_{mk}) - \delta$

Theorem 7.4.1 *Consider the Broyden-Fletcher-Goldfarb-Shanno method (i.e., the Broyden method with $\phi \equiv 1$) without restarts and assume that the method converges to a nondegenerate local minimizer x^* of a three times continuously differentiable function f. Then the method converges to x^* superlinearly.*

This result has also extensions onto the "practical" – with properly specified inexact line search – versions of the method.

7.4.4.3 Appendix: derivation of the BFGS updating formula

To get the starting point for our developments, note that relation (7.4.7), i.e.,

$$S_{t+1}q_t = p_t,$$

for the case of symmetric positive definite S_{t+1}, is equivalent to the relation

$$H_{t+1}p_t = q_t \tag{7.4.15}$$

with symmetric positive definite $H_{t+1} \equiv S_{t+1}^{-1}$. Thus,

each policy $\mathcal{P}$ of generating positive definite symmetric matrices S_t maintaining (7.4.7) induces certain policy $\mathcal{P}^*$ of generating positive definite symmetric matrices $H_t = S_t^{-1}$ maintaining (7.4.15),

and, of course, vice versa:

each policy $\mathcal{P}^*$ of generating positive definite symmetric matrices H_t maintaining (7.4.15) induces certain policy of generating positive definite symmetric matrices $S_t = H_t^{-1}$ maintaining (7.4.7).

Now, we know how to generate matrices H_t satisfying relation (7.4.15) – this is, basically, the same problem as we already have studied, but with swapped q_t and p_t. Thus,

given any one of the above updating formulae for S_t, we can construct a "complementary" updating formula for H_t by replacing, in the initial formula, all S's by H's and interchanging q's and p's.

For example, the complementary formula for the DFP updating scheme (7.4.8) is

$$H_{t+1} = H_t + \frac{1}{q_t^T p_t} q_t q_t^T - \frac{1}{p_t^T H_t p_t} H_t p_t p_t^T H_t, \tag{7.4.16}$$

– it is called the Broyden-Fletcher-Goldfarb-Shanno updating of H_t.

We have the following analogy to Proposition 7.4.2:

Proposition 7.4.3 *Let H_t be positive definite symmetric matrix, let x_{t-1} be an arbitrary point with $g_{t-1} = \nabla f(x_{t-1}) \neq 0$, and let d_t be an arbitrary descent direction of f at x_{t-1} (i.e., $d_t^T g_{t-1} < 0$). Let, further, x_t be the minimizer of f on the ray $\{x_{t-1} + \gamma d_t \mid \gamma \geq 0\}$, and let*

$$p_t = x_t - x_{t-1},\ q_t = \nabla f(x_t) - \nabla f(x_{t-1}) \equiv g_t - g_{t-1}.$$

Then the matrix H_{t+1} given by (7.4.16) is symmetric positive definite and satisfies (7.4.15).

Proof. From the proof of Proposition 7.4.2 we know that $p_t^T q_t > 0$, and it remains to apply Proposition 7.4.1 to the data $R = H_t$, $p = q_t$, $q = p_t$. ∎

According to Proposition 7.4.3, we can look at (7.4.16) as at certain maintaining (7.4.15) policy $\mathcal{P}^*$ of updating positive definite symmetric matrices H_t. As we know, every policy of this type induces certain policy of updating positive definite matrices $S_t = H_t^{-1}$ which maintains (7.4.7). In our case the induced policy is given by

$$S_{t+1} = \left[S_t^{-1} + \frac{1}{p_t^T q_t} q_t q_t^T - \frac{1}{p_t^T S_t^{-1} p_t} S_t^{-1} p_t p_t^T S_t^{-1} \right]^{-1}. \tag{7.4.17}$$

It can be shown by a direct computation (which I skip), that the latter relation is nothing but the BFGS updating formula.

Lecture 8

Convex Programming

Today we start the last part of the Course – methods for solving *constrained* optimization problems. This lecture is in sense slightly special – we shall speak about *Convex Programming problems*. As we remember, a Convex Programming problem is the one of the form

$$\min_x \{f_0(x) : f_i(x) \leq 0,\, i = 1, ..., m,\ x \in G \subset \mathbf{R}^n\}, \tag{8.0.1}$$

where

- G – the *domain* of the problem – is a closed *convex* set in $\mathbf{R}^n$;
- the objective f_0 and the constraints f_i, $i = 1, ..., m$, are *convex* functions on $\mathbf{R}^n$; for the sake of simplicity, from now on we assume that the domains of these functions are the entire $\mathbf{R}^n$.

Convex Programming is, in a sense, the "solvable case" in Nonlinear Optimization: as we shall see in a while, convex programs, in contrast to the general ones, can be solved efficiently: one can approximate their *global* solutions by *globally linearly converging, with data-independent ratio*, methods. This phenomenon – possibility to approximate efficiently global solutions – has no analogy in the general nonconvex case[1)] and comes from nice geometry of convex programs.

Computational tools for Convex Programming is the most developed, both theoretically and algorithmically, area in Continuous Optimization, and what follows is nothing but a brief introduction to this rich area. In fact I shall speak about only one method – the *Ellipsoid algorithm*, since it is the simplest way to achieve the main goal of the lecture – to demonstrate that Convex Programming indeed is "solvable". Practical conclusion of this theoretical in its essence phenomenon is, in my opinion, very important, and I would formulate it as follows:

When modelling a real-world situation as an optimization program, do your best to make the program convex; I strongly believe that it is more important than to minimize the number of design variables or to ensure smoothness of the objective and the constraints. Whenever you are able to cast the problem as a convex program, you get in your disposal wide spectrum of efficient and reliable optimization tools.

8.1 Preliminaries

Let me start with recalling to you two important notions of Convex Analysis – those of *a subgradient* of a convex function and *a separating plane*.

[1)] Recall that in all our previous considerations the best we could hope for was the convergence to a local minimizer of the objective, and we never could guarantee this local minimizer to be the global one. Besides this, basically all our rate-of-convergence results were asymptotical and depended on local properties of the objective

8.1.1 Subgradients of convex functions

In our previous considerations, we always required the objective to be smooth – at least once continuously differentiable – and used the first order information (the gradients) or the second order one (the gradients and the Hessians) of the objective. In Convex Programming, there is no necessity to insist on differentiability of the objective and the constraints, and the role of gradients is played by *subgradients*. Recall that if g is a convex function on $\mathbf{R}^n$, then at each point $x \in \mathbf{R}^n$ there exists at least one *subgradient* $g'(x)$ of g – vector such that

$$g(y) \geq g(x) + (y-x)^T g'(x). \tag{8.1.1}$$

Geometrically: the graph of the linear function

$$y \mapsto g(x) + (y-x)^T g'(x)$$

is everywhere below the graph of g itself, and at the point x both graphs touch each other. Generally speaking, subgradient of g at a point x is not unique; it is unique if and only if g is differentiable at x, and in this case the unique subgradient of g at x is exactly the gradient $\nabla g(x)$ of g at x.

When speaking about methods for solving (8.0.1), we always assume that we have in our disposal a *First order oracle*, i.e., a routine which, given on input a point $x \in \mathbf{R}^n$, returns on output the values $f_i(x)$ and certain subgradients $f_i'(x)$, $i = 0, ..., m$ of the objective and the constraints at x.

8.1.2 Separating planes

Given a closed convex set $G \subset \mathbf{R}^n$ and a point x outside G, one can always *separate x from G by a hyperplane*, i.e., to point out a *separator* – vector e such that

$$e^T x > \sup_{y \in G} e^T y. \tag{8.1.2}$$

Geometrically: whenever $x \notin G$ (G is convex and closed), we can point out a *hyperplane*

$$\Pi = \{y \mid e^T y = a\}$$

such that x and G belong to opposite open half-spaces into which Π splits $\mathbf{R}^n$:

$$e^T x > a \ \& \ \sup_{y \in G} e^T y < a.$$

When speaking about methods for solving (8.0.1), we always assume that we have in our disposal a *Separation oracle* for G, i.e., a routine which, given on input a point $x \in \mathbf{R}^n$, reports whether $x \in G$, and if it is not the case, returns "the proof" of the relation $x \notin G$, i.e., a vector e satisfying (8.1.2).

Remark 8.1.1 Implementation of the oracles. The assumption that when solving (8.0.1), we have in our disposal a First order oracle for the objective and the constraints and a Separation oracle for the domain G are crucial for our abilities to solve (8.0.1) efficiently. Fortunately, this assumption indeed is valid in all "normal" cases. Indeed, typical convex functions f arising in applications are either differentiable – and then, as in was already mentioned, subgradients are the same as gradients, so that they are available whenever we have an "explicit" representation of the function – or are upper bounds of differentiable convex functions:

$$f(x) = \sup_{\alpha \in A} f_\alpha(x).$$

In the latter case we have no problems with computing subgradients of f in the *discrete case* – when the index set A is finite. Indeed, it is immediately seen from the definition of a subgradient that if $\alpha(x)$ is the index of the largest at x (or a largest, if there are several of them) of the functions f_α, then the gradient of $f_{\alpha(x)}$ at the point x is a subgradient of f as well. Thus, if all f_α are "explicitly given", we can simply compute all of them at x and choose the/a one largest at the point; its value will be the value of f at x, and its gradient at the point will be a subgradient of f at x.

Now, the domain G of (8.0.1) typically is given by a set of convex inequalities:

$$G = \{x \in \mathbf{R}^n \mid g_j(x) \leq 0,\ j = 1, ..., k\}$$

with explicitly given and simple – differentiable – convex functions g_j, e.g.

$$G = \{x \mid (x-a)^T(x-a) - R^2 \leq 0\}$$

(centered at a Euclidean ball of radius R) or

$$G = \{x \mid a_i \leq x \leq b_i,\ i = 1, ..., n\}$$

(a box). Whenever it is the case, we have no problems with Separation oracle for G: given x, it suffices to check whether all the inequalities describing G are satisfied at x. If it is the case, then $x \in G$, otherwise $g_i(x) > 0$ for some i, and it is immediately seen that $\nabla g_i(x)$ can be chosen as separator.

It should be stressed, anyhow, that not all convex programs admit "efficient" oracles. For example, we may meet with a problem where the objective (or a constraint) is given by "continuous" maximization like

$$f(x) = \max_{\alpha \in [0,1]} f_\alpha(x)$$

("semi-infinite programs"); this situation occurs in the problems of best uniform approximation of a given function $\phi(\alpha)$, $0 \leq \alpha \leq 1$, by a linear combination $\sum_{i=1}^{n} x_i \phi_i(\alpha)$ of other given functions; the problem can be written down as a "simple" convex program

$$\min_x \left\{ f(x) \equiv \max_{\alpha \in [0,1]} |\phi(\alpha) - \sum_{i=1}^{n} x_i \phi_i(\alpha)| \right\};$$

this is a convex program, but whether it can or cannot be solved efficiently, it heavily depends on whether we know how to perform efficiently maximization in α required to compute the value of f at a point.

8.2 The Ellipsoid Method

We are about to present one of the most general algorithms for convex optimization – the *Ellipsoid method*. The idea of the method is more clear when we restrict ourselves with the problem of the type

$$\min_x \{f(x) : x \in G \subset \mathbf{R}^n\} \tag{8.2.1}$$

of minimizing a convex function f on a *solid* G, i.e., a closed and bounded convex set with a nonempty interior. In the mean time we shall see that general problem (8.0.1) can be easily reduced to (8.2.1).

8.2.1 The idea

The method extends onto multi-dimensional case the scheme of the simplest method for univariate minimization, namely, of the Bisection (Lecture 2). The idea is very simple: let us take an arbitrary interior point $x_1 \in G$ and compute a subgradient $f'(x_1)$ of the objective at the point, so that

$$f(x) - f(x_1) \geq (x - x_1)^T f'(x_1). \tag{8.2.2}$$

It is possible that $f'(x_1) = 0$; then (8.2.2) says that x_1 is a global minimizer of f on $\mathbf{R}^n$, and since this global minimizer belongs to G, it is an optimal solution to the problem, and we can terminate. Now assume that $f'(x_1) \neq 0$ and let us ask ourselves what can be said about localization of the optimal set, let it be called X^*. The answer is immediate:

from (8.2.2) it follows that if x belongs to the open half-space

$$\Pi_1^+ = \{x \mid (x - x_1)^T f'(x_1) > 0\},$$

so that the right hand side in (8.2.2) is positive at x, then x for sure is not optimal: (8.2.2) says that $f(x) > f(x_1)$, so that x is worse than feasible solution x_1. Consequently, the optimal set (about which we initially knew only that $X^* \subset G$) belongs to the new localizer

$$G_1 = \{x \in G \mid (x - x_1)^T f'(x_1) \leq 0\}.$$

This localizer again is a convex solid (as the intersection of G and a closed half-space with the boundary passing through the interior point x_1 of G) *and is smaller than G* (since an interior point x_1 of G is a boundary point of G_1).

Thus, choosing somehow an interior point x_1 in the "initial localizer of the optimal set" $G \equiv G_0$ and looking at $f'(x_1)$, we either terminate with exact solution, or can perform a cut – pass from G_0 to a smaller solid G_1 which also contains the optimal set. In this latter case we can iterate the construction – choose somehow an interior point x_2 in G_1, compute $f'(x_2)$ and, in the case of $f'(x_2) \neq 0$, perform a new cut – replace G_1 with the new localizer

$$G_2 = \{x \in G_1 \mid (x - x_2)^T f'(x_2) \leq 0\},$$

and so on. With the resulting recurrence, we either terminate at certain step with exact solution, or generate a sequence of shrinking solids

$$G = G_0 \supset G_1 \supset G_2 \supset ...,$$

every of the solids containing the optimal set.

It can be guessed that if x_t are chosen properly, then the localizers G_t "shrink to X^* at certain reasonable rate", and the method converges. This is, e.g., the case with Bisection: there all G_t are segments (what else could be a solid on the axis?), and x_t is chosen as the center of G_{t-1}; as a result, the lengths of the segments G_t go to 0 linearly with the ratio $\frac{1}{2}$, and since all the localizers contain X^*, the method converges at the same linear rate.

In multi-dimensional case the situation is much more difficult:

- the sets G_t can be more or less arbitrary solids, and it is not clear what does it mean a "center" of G_t, i.e., how to choose $x_{t+1} \in G_t$ in order to achieve "significant shrinkage" of the localizer at a step (by the way, how to measure this shrinkage?) On the other hand, it is clear that if x_{t+1} is badly chosen (e.g., is close to the boundary of G_t) and the new cut is "badly oriented", the next localizer G_{t+1} may be "almost as large as G_t";
- in the one-dimensional case the linear sizes of the localizers tend to 0 linearly with ratio $\frac{1}{2}$, and this is the source of convergence of x_t to the minimizer of f. It is absolutely clear that in the multi-dimensional case we cannot enforce the linear sizes of G_t to go to 0 (look what happens if $G = G_0$ is the unit square on the plane and f does not depend on the first coordinate; in this case all localizers will be rectangles of the same width, and consequently we cannot enforce their linear sizes to go to 0). Thus, we should think carefully how to measure sizes of the localizers – with bad choice of the notion of size, we are unable to push it to 0.

The above remarks demonstrate that it is not that straightforward – to extend Bisection onto the multi-dimensional case. Nevertheless, there are satisfactory ways to do it.

8.2.2 The Center-of-Gravity method

The first which comes to mind is to measure the sizes of the localizers as their n-dimensional volumes $\mathrm{Vol}(G_t)$ and to use, as x_{t+1}, the *center of gravity* of G_t:

$$x_{t+1} = \frac{1}{\mathrm{Vol}(G_t)} \int_{G_t} x dx.$$

A (very nontrivial) geometrical fact is that with this *Center of Gravity* policy we get linear convergence of the volumes of G_t to 0:

$$\mathrm{Vol}(G_t) \le \left[1 - \left(\frac{n}{n+1}\right)^n\right]^t \mathrm{Vol}(G_0),$$

which in turn implies linear convergence in terms of the residual in the objective value: if x^t is the best – with the smallest value of the objective – of the points $x_1, ..., x_t$, then

$$f(x^t) - \min_G f \le \left[1 - \left(\frac{n}{n+1}\right)^n\right]^{t/n} [\max_G f - \min_G f]. \tag{8.2.3}$$

(8.2.3) demonstrates *global linear convergence of the Center-of-Gravity method with objective-independent convergence ratio*

$$\kappa(n) = \left[1 - \left(\frac{n}{n+1}\right)^n\right]^{1/n} \le (1 - \exp\{-1\})^{1/n}.$$

Consequently, *to get an ϵ-solution to the problem – to find a point $x^t \in G$ with*

$$f(x^t) - \min_G f \le \epsilon[\max_G f - \min_G f]$$

– it requires to perform no more than

$$\rfloor \frac{\ln \frac{1}{\epsilon}}{\ln \frac{1}{\kappa(n)}} \lfloor \le 2.13 n \ln \frac{1}{\epsilon} + 1 \tag{8.2.4}$$

steps of the method[2)].

Look how strong is the result: our *global* efficiency estimate is objective-independent: no condition numbers (and even no smoothness assumptions at all) are involved! f may be nonsmooth, may possess all kinds of degeneracy, etc., and all this does not influence the estimate!

Let me add that the Center-of-Gravity method is, in certain precise sense, an optimal method for convex optimization.

A weak point of the method is the necessity to find, at every step, the center of gravity of the previous localizer. To find the center of gravity of a general type multi-dimensional solid (and even of a general type polytope) is, computationally, an extremely difficult problem. Of course, if G (and then every G_t) is a polytope, this problem is "algorithmically solvable" – one can easily point out a method which solves the problem in finitely many arithmetic operations. Unfortunately, the number of arithmetic operations required by all known methods for finding the center of gravity grows exponentially with n, so that are unable to compute centers of gravity in reasonable time already in the dimension 5-10. As a result, the Center-of-Gravity method is of "academic interest" only – it cannot be used as a computational tool.

8.2.3 From Center-of-Gravity to the Ellipsoid method

The Ellipsoid method can be viewed as certain "computationally tractable" approximation to the Center-of-Gravity method. The idea is to enforce all the localizers to have "nice geometry" convenient for finding the centers of gravity, namely, to be ellipsoids. Assume, e.g., that $G_0 = G$ is an ellipsoid. Then there is no difficulty to point out the center of G_0, so that we have no problems with the first step of the Center-of-Gravity method. Unfortunately, the resulting localizer, let it be called G_1^+, will be not an ellipsoid, but a "half-ellipsoid" – the intersection of ellipsoid G_0 and a half-space with the boundary hyperplane passing through the center of G_0; this solid is not that convenient for finding the center of gravity. To restore the geometry of the localizer, let us cover the half-ellipsoid G_1^+ by the ellipsoid of the smallest volume G_1 containing G_1^+, and let us take this G_1 as our new localizer. Note that G_1 indeed is a localizer of the optimal set X^* (since the latter is contained already in G_1^+, and G_1 covers G_1^+). Now we are in the same situation as at the very beginning, but with G_0 replaced with G_1, and can iterate the construction – to

[2]here and in what follows we use the standard notation $\rfloor a \lfloor$ to denote the smallest integer $\ge$ a real a

take the center x_2 of the ellipsoid G_1, to perform a new cut, getting a new half-ellipsoid G_1^+ which covers the optimal set, to embed it into the ellipsoid G_2 of the smallest possible volume, etc. In this scheme, we actually make a kind of trade-off between efficiency of the routine and computational complexity of a step: when extending the "actual localizers" – half-ellipsoids – to ellipsoids, we add points which for sure could not be optimal solutions, and thus slow the procedure down. At the cost of this slowing the process down we, anyhow, enable ourselves to deal with "simple sets" – ellipsoids, and thus reduce dramatically the computational complexity of a step.

There are two points which should be clarified.

- first, it is unclear in advance whether we indeed are able to decrease at linear rate the volumes of sequential localizers and thus get a converging method – it could happen that, when extending the half-ellipsoid G_{t+1}^+ to the ellipsoid G_{t+1} of the smallest volume containing G_{t+1}^+, we come back to the previous ellipsoid G_t, so that no progress in volume is achieved. Fortunately, this is <u>not</u> the case: the procedure reduces the volumes of the sequential ellipsoids G_t by factor $\kappa^*(n) \leq \exp\{-\frac{1}{2n}\}$, thus enforcing the volumes of G_t to go to 0 at linear rate with the ratio $\kappa^*(n)$ [3]. This ratio is worse than the one for the Center-of-Gravity method (there the ratio was at most absolute constant $1 - \exp\{-1\}$, now it is dimension-dependent constant close to $1 - \frac{1}{2n}$); but we still have linear convergence!

- second, it was assumed that G is an ellipsoid. What to do if it is not so? The answer is easy: let us choose, as G_0, an arbitrary ellipsoid which covers the domain G of the problem; such a G_0 for sure will be a localizer of the optimal set, and this is all we need. This answer is good, but there is a weak point: it may happen that the center x_1 of the ellipsoid G_0 is outside G; how should we perform the first cut in this case? Moreover, this "bad" situation – when the center x_{t+1} of the current localizer G_t is outside G – may eventually occur even when $G_0 = G$ is an ellipsoid: at each step of the method, we add something to the "half" of the previous localizer, and this something could contain points not belonging to G. As a result, G_t, generally speaking, is not contained in G, and it may happen that the center of G_t is outside G. And to the moment we know how to perform cuts only through points of G, not through arbitrary points of $\mathbf{R}^n$.

  We can immediately resolve the indicated difficulty. Given the previous ellipsoidal localizer G_t and its center x_{t+1}, let us ask the Separation oracle whether $x_{t+1} \in G$. If it is the case, we have no problems with the cut – we call the First order oracle, get a subgradient of f at x_{t+1} and use it as it was explained above to produce the cut. Now, if x_{t+1} is not in G, the Separation oracle will return a separator e:

$$e^T x_{t+1} > \max_{x \in G} e^T x.$$

  Consequently, all the points $x \in G$ satisfy the inequality

$$(x - x_{t+1})^T e < 0,$$

  and we can use e for the cut, setting

$$G_{t+1}^+ = \{x \in G_t \mid (x - x_{t+1})^T e \leq 0\}.$$

  Since the inequality which "cuts G_{t+1}^+ off G_t" is satisfied on the entire G and, consequently, on X^*, and the latter set was assumed to belong to G_t (G_t is a localizer!), we conclude that $X^* \subset G_{t+1}^+$, and this is all we need.

After all explanations and remarks, we can pass to formal description of the Ellipsoid method.

[3] The fact that even after extending G_{t+1}^+ to G_{t+1} we still have progress in volume heavily depends on the specific geometrical properties of ellipsoids; if. e.g., we would try to replace the ellipsoids with boxes, we would fail to ensure the desired progress. The ellipsoids, anyhow, are not the only solids appropriate for our goals; we could use simplexes as well, although with worse progress in volume per step

8.2.4 The Algorithm

8.2.4.1 How to represent an ellipsoid

An ellipsoid is a geometrical entity; to run an algorithm, we should deal with numerical representations of these entities. The most convenient for our goals is to represent an ellipsoid as the image of the unit Euclidean ball under nonsingular affine mapping:

$$E = E(c, B) = \{x = c + Bu \mid u^T u \leq 1\}. \tag{8.2.5}$$

Here $c \in \mathbf{R}^n$ is the center of the ellipsoid and B is an $n \times n$ nonsingular matrix. Thus, to say "an ellipsoid" in what follows means exactly to say "the set (8.2.5) given by a pair (c, B) comprised of vector $c \in \mathbf{R}^n$ and a nonsingular $n \times n$ matrix B"; you may think about this representation as about the *definition* of an ellipsoid.

The volume the ellipsoid $E(c, b)$ is

$$\mathrm{Vol}(E(c, B)) = |\mathrm{Det}\ B| v(n),$$

$v(n)$ being the volume of the unit n-dimensional Euclidean ball V_n; indeed, $E(c, B)$ is the image of V_n under affine mapping, B being the matrix of the homogeneous part of the mapping, and such a transformation multiplies the volumes by $|\mathrm{Det}\ B|$.

We need the following simple

Lemma 8.2.1 *Let $n > 1$, let*

$$E = E(c, B) = \{x = c + Bu \mid u^T u \leq 1\}$$

be an ellipsoid in $\mathbf{R}^n$, and let

$$E^+ = \{x \in E \mid (x - c)^T e \leq 0\} \quad [e \neq 0]$$

be a "half-ellipsoid" – the intersection of E and a half-space with the boundary hyperplane passing through the center of E. Then E^+ can be covered by ellipsoid $E' = E(c', B')$ of the volume

$$\mathrm{Vol}(E') = \kappa^*(n)\, \mathrm{Vol}(E),$$

$$\kappa^*(n) = \frac{n^2}{n^2 - 1}\sqrt{\frac{n-1}{n+1}} \leq \exp\{-\frac{1}{2(n-1)}\}. \tag{8.2.6}$$

The parameters c' and B' of the ellipsoid E' are given by

$$B' = \alpha(n) B - \gamma(n)(Bp)p^T, \quad c' = c - \frac{1}{n+1} Bp, \tag{8.2.7}$$

where

$$\alpha(n) = \left\{\frac{n^2}{n^2 - 1}\right\}^{1/4}, \quad \gamma(n) = \alpha(n)\sqrt{\frac{n-1}{n+1}}, \quad p = \frac{B^T e}{\sqrt{e^T B B^T e}}.$$

To prove the lemma, it suffices to reduce the situation to the similar one with E being the unit Euclidean ball $V = V_n$; indeed, since E is the image of V under the affine transformation $u \mapsto Bu + c$, the half-ellipsoid E^+ is the image, under this transformation, of the half-ball

$$V^+ = \{u \in V \mid (B^T e)^T u \leq 0\} = \{u \in V \mid p^T u \leq 0\}.$$

Now, it is quite straightforward to verify that a half-ball indeed can be covered by an ellipsoid V' with the volume being the required fraction of the volume of V; you may take

$$V' = \{x \mid (x + \frac{1}{n+1} p)^T [\alpha(n) I_n - \gamma(n) p p^T]^{-2} (x + \frac{1}{n+1} p) \leq 1\}.$$

It remains to note that the image of V' under the affine transformation which maps the unit ball V onto the ellipsoid E is an ellipsoid which clearly contains the half-ellipsoid E^+ and is in the same ratio of volumes with respect to E as V' is with respect to the unit ball V (since the ratio of volumes remains invariant under affine transformations). The ellipsoid E' given in formulation of the lemma is nothing but the image of the above V' under our affine transformation. ∎

8.2.4.2 The Ellipsoid algorithm

The algorithm is as follows

Algorithm 8.2.1 [The Ellipsoid algorithm for convex program $\min_x \{f(x) : x \in G \subset \mathbf{R}^n\}$]
Assumptions:

- *G is a solid (bounded and closed convex set with a nonempty interior)*
- *we are given First order oracle for f and Separation oracle for G*
- *we are able to point out an ellipsoid $G_0 = E(c_0, B_0)$ which contains G.*

Initialization: *set $t = 1$*
Step t: Given ellipsoid $G_{t-1} = E(c_{t-1}, B_{t-1})$, set $x_t = c_{t-1}$ and act as follows:
1) [Call the Separation Oracle] *Call the Separation oracle, x_t being the input. If the oracle says that $x_t \in G$, call the step t productive and go to 2), otherwise call the step t non-productive, set e_t equal to the separator returned by the oracle:*

$$x \in G \Rightarrow (x - x_t)^T e_t < 0$$

and go to 3)
2) [Call the First order oracle] *Call the First order oracle, x_t being the input. If $f'(x_t) = 0$, terminate – $x_t \in G$ is the minimizer of f on the entire $\mathbf{R}^n$ (see the definition (8.1.1) of a subgradient) and is therefore an optimal solution to the problem. In the case of $f'(x_t) \neq 0$ set*

$$e_t = f'(x_t)$$

and go to 3).
3) [Update the ellipsoid] *Update the ellipsoid $E(c_{t-1}, B_{t-1})$ into $E(c_t, B_t)$ according to formula (8.2.7) with $e = e_t$, i.e., set*

$$B_t = \alpha(n)B_{t-1} - \gamma(n)(B_{t-1}p_t)p_t^T, \quad c_t = c_{t-1} - \frac{1}{n+1}B_{t-1}p_{t-1},$$

where

$$\alpha(n) = \left\{\frac{n^2}{n^2-1}\right\}^{1/4}, \quad \gamma(n) = \alpha(n)\sqrt{\frac{n-1}{n+1}}, \quad p_t = \frac{B_{t-1}^T e_t}{\sqrt{e_t^T B_{t-1} B_{t-1}^T e_t}}.$$

Replace t with $t+1$ and go to the next step.
Approximate solution x^t generated by the method after t steps is, by definition, the best (with the smallest value of f_0) of the points x_j generated at the productive steps $j \leq t$ [if the steps $1, ..., t$ are non-productive, x^t is undefined].

8.2.5 The Ellipsoid algorithm: rate of convergence

We are about to prove the main result on the Ellipsoid method.

Theorem 8.2.1 [Rate of convergence of the Ellipsoid algorithm]
Let $n > 1$, and let convex program (8.2.1) satisfying the assumptions from the description of the Ellipsoid algorithm 8.2.1 be solved by the algorithm. Let

$$N(\epsilon) = \lfloor 2n(n-1)\ln\left(\frac{\mathcal{V}}{\epsilon}\right)\rfloor,$$

where $0 < \epsilon < 1$ and

$$\mathcal{V} = \left[\frac{\mathrm{Vol}(E(c_0, B_0))}{\mathrm{Vol}(G)}\right]^{1/n}$$

Then, for any $\epsilon \in (0,1)$, the approximate solution $x^{N(\epsilon)}$ found by the method in course of the first $N(\epsilon)$ steps, is well-defined and is an ϵ-solution to the problem, i.e., belongs to G and satisfies the inequality

$$f(x^{N(\epsilon)}) - \min_G f \le \epsilon[\max_G f - \min_G f]. \tag{8.2.8}$$

Proof. For the sake of brevity, let $N = N(\epsilon)$. We may assume that the method does not terminate in course of the first N steps – otherwise there is nothing to prove: the only possibility for the method to terminate is to find an exact optimal solution to the problem.

Let us fix $\epsilon' \in (\epsilon, 1)$, and let x^* be an optimal solution to (8.2.1) (it exists, since the domain of the problem is compact and the objective, being convex on $\mathbf{R}^n$, is continuous (we had such a theorem in the course Optimization I) and therefore attains its minimum on compact set G.

1^0. Set

$$G^* = x^* + \epsilon'(G - x^*),$$

so that G^* is the image of G under homothety transformation with the center at x^* and the coefficient ϵ'.

By construction, we have

$$\mathrm{Vol}(G^*) = (\epsilon')^n \,\mathrm{Vol}(G) > \epsilon^n \,\mathrm{Vol}(G) \tag{8.2.9}$$

("homothety with coefficient $\alpha > 0$ in $\mathbf{R}^n$ multiplies volumes by α^n"). On the other hand, by Lemma 8.2.1 we have $\mathrm{Vol}(E(c_t, B_t)) \le \exp\{-1/(2(n-1))\}\,\mathrm{Vol}(E(c_{t-1}, B_{t-1}))$, whence

$$\mathrm{Vol}(E(c_N, B_N) \le \exp\{-\frac{N}{2(n-1)}\}\,\mathrm{Vol}(E(c_0, B_0)) \le$$

[definition of $N = N(\epsilon)$]

$$\le \mathcal{V}^{-N}\epsilon^N \,\mathrm{Vol}(E(c_0, B_0)) =$$

[definition of $\mathcal{V}$]

$$= \frac{\mathrm{Vol}(G)}{\mathrm{Vol}(E(c_0, B_0))}\epsilon^N \,\mathrm{Vol}(E(c_0, B_0)) = \epsilon^N \,\mathrm{Vol}(G).$$

Comparing the resulting inequality with (8.2.9), we conclude that $\mathrm{Vol}(E(c_N, B_N)) < \mathrm{Vol}(G^*)$, so that

$$G^* \backslash E(c_N, B_N) \ne \emptyset. \tag{8.2.10}$$

2^0. According to (8.2.10), there exists

$$y \in \mathrm{Vol}(G^*) \backslash E(c_N, B_N).$$

I claim that

$$y = (1-\epsilon')x^* + \epsilon' z \text{ for some } z \in G; \qquad (y - x_t)^T e_t > 0 \text{ for some } t \le N \tag{8.2.11}$$

The first relation in (8.2.11) comes from the inclusion $y \in G^*$: by definition,

$$G^* = x^* + \epsilon'(G - x^*) = \{x^* + \epsilon'(z - x^*) \mid z \in G\}.$$

To prove the second relation in (8.2.11), note that from the first one $y \in G \subset E(c_0, B_0)$, while by the origin of y we have $y \notin E(c_N, B_N)$. Consequently, there exists $t \le N$ such that

$$y \in E(c_{t-1}, B_{t-1}) \;\&\; y \notin E(c_t, B_t). \tag{8.2.12}$$

According to our policy of updating the ellipsoids (see Lemma 8.2.1), $E(c_t, B_t)$ contains the half-ellipsoid

$$E^+(c_{t-1}, B_{t-1}) = \{x \in E(c_{t-1}, B_{t-1}) \mid (x - x_t)^T e_t \le 0\};$$

this inclusion and (8.2.12) demonstrate that $(y - x_t)^T e_t > 0$, as required in the second relation in (8.2.11).

3^0. Now let us look what happens at the step t given by the second relation in (8.2.11). First of all, I claim that the step t is productive: $x_t \in G$. Indeed, otherwise, by construction of the method, e_t would be a separator of x_t and G:

$$(x - x_t)^T e_t < 0 \quad \forall x \in G,$$

but this relation, as we know from (8.2.11), is violated at $y \in G$ and therefore cannot take place.

Thus, t is productive, whence, by construction of the method, $e_t = f'(x_t)$. Now the second relation in (8.2.11) reads

$$(y - x_t)^T f'(x_t) > 0,$$

whence, by definition of subgradient, $f(y) > f(x_t)$. This inequality, along with productivity of the step t and the definition of approximate solutions, says that x^N is well-defined and

$$f(x^N) \leq f(y). \tag{8.2.13}$$

4^0. Now we are done. By the first relation in (8.2.11) and due to convexity of f we have

$$f(y) \leq (1 - \epsilon')f(x^*) + \epsilon' f(z),$$

whence

$$f(y) - \min_G f \equiv f(y) - f(x^*) \leq \epsilon'(f(z) - f(x^*)) \leq \epsilon'[\max_G f - \min_g f] \qquad [z \in G\ !];$$

combining the resulting inequality with (8.2.13), we get

$$f(x^N) - \min_G f \leq \epsilon'[\max_G f - \min_G f].$$

The latter inequality is valid for all $\epsilon' \in (\epsilon, 1)$, and (8.2.8) follows. ∎

8.2.6 Ellipsoid method for problems with functional constraints

To the moment we spoke about the Ellipsoid method as applied to problem (8.2.1) without functional constraints. There is no difficulty to extend the method onto general convex problems (8.0.1). To this end it suffices to note that (8.0.1) can be immediately rewritten as problem of the type (8.2.1), namely, as

$$\min_x \left\{ f(x) \equiv f_0(x) : x \in \hat{G} \equiv \{x \in G \mid\ f_i(x) \leq 0,\ i = 1, ..., m\} \right\}. \tag{8.2.14}$$

All we need to solve this latter problem by the Ellipsoid algorithm is

- to equip (8.2.14) by the Separation and the First order oracles

  this is immediate: the First order oracle for (8.2.14) is readily given by the one for (8.0.1). The Separation oracle for $\hat{G}$ can be immediately constructed from the Separation oracle for G and the First order oracle for (8.0.1): given $x \in \mathbf{R}^n$, we first check whether the point belongs to G, If it is not the case, then it, of course, does not belong to $\hat{G}$ as well, and the separator (given by the Separation oracle for G on the input x) separates x from $\hat{G}$. Now, if $x \in G$, we can ask the First order oracle about the values and subgradients of $f_1, ..., f_m$ at x. If all the values are nonpositive, then $x \in \hat{G}$; if one of this values is positive, then $x \notin \hat{G}$ and, moreover, the vector $f_i'(x)$ ($i \geq 1$ is such that $f_i(x) > 0$) can be used as a separator of x and $\hat{G}$ (Remark 8.1.1)

- to ensure that $\hat{G}$ is a solid

  (we shall simply assume this).

The resulting algorithm is as follows

Algorithm 8.2.2 [The Ellipsoid algorithm for convex program (8.0.1)]
Assumptions:

- $\hat{G} = \{x \in G \mid f_i(x) \leq 0, i = 1, ..., m\}$ *is a solid (bounded and closed convex set with a nonempty interior)*
- *we are given First-order oracle for* $f_0, ..., f_m$ *and Separation oracle for* G
- *we are able to point out an ellipsoid* $G_0 = E(c_0, B_0)$ *which contains* $\hat{G}$.

Initialization: set $t = 1$
Step t*: Given ellipsoid* $G_{t-1} = E(c_{t-1}, B_{t-1})$*, set* $x_t = c_{t-1}$ *and act as follows:*

1) [Call the Separation oracle] *Call the Separation oracle for* G*,* x_t *being the input. If the oracle says that* $x_t \in G$*, go to 2), otherwise call the step* t *non-productive, set* e_t *equal to the separator returned by the oracle:*

$$x \in G \Rightarrow (x - x_t)^T e_t < 0$$

and go to 4)

2) [Call the First order oracle] *Call the First order oracle,* x_t *being the input, and check whether* $f_i(x_t) \leq 0$*,* $i = 1, ..., m$*. If* $f_i(x_t) \leq 0$ *for all* $i \geq 1$*, call the step* t *productive and look at* $f_0'(x_t)$*. If* $f_0'(x_t) = 0$*, terminate – *x_t *is feasible for (8.0.1) and is the minimizer of* f *on the entire* $\mathbf{R}^n$ *(see the definition (8.1.1) of a subgradient), whence it is an optimal solution to the problem. In the case of* $f_0'(x_t) \neq 0$ *set*

$$e_t = f_0'(x_t)$$

and go to 4).

3) [The case of $x_t \in G$ and $f_i(x_t) > 0$ for some $i \geq 1$] *Call step* t *non-productive and find* $i \geq 1$ *such that* $f_i(x_t) > 0$ *(when we arrive at 3), such an* i *exists), set*

$$e_t = f_i'(x_t)$$

and go to 4).

4) [Updating the ellipsoid] *Update the ellipsoid* $E(c_{t-1}, B_{t-1})$ *into* $E(c_t, B_t)$ *according to formula (8.2.7) with* $e = e_t$*, i.e., set*

$$B_t = \alpha(n)B_{t-1} - \gamma(n)(B_{t-1}p_t)p_t^T, \quad c_t = c_{t-1} - \frac{1}{n+1}B_{t-1}p_{t-1},$$

where

$$\alpha(n) = \left\{\frac{n^2}{n^2-1}\right\}^{1/4}, \quad \gamma(n) = \alpha(n)\sqrt{\frac{n-1}{n+1}}, \quad p_t = \frac{B_{t-1}^T e_t}{\sqrt{e_t^T B_{t-1} B_{t-1}^T e_t}}.$$

Replace t *with* $t+1$ *and go to the next step.*

Approximate solution x^t *generated by the method after* t *steps is, by definition, the best (with the smallest value of* f*) of the points* x_j *generated at the productive steps* $j \leq t$ *[if the steps* $1, ..., t$ *are non-productive,* x^t *is undefined].*

The following result is an immediate corollary of Theorem 8.2.1.

Theorem 8.2.2 [Rate of convergence of the Ellipsoid algorithm on problems with functional constraints] *Let* $n > 1$*, and let convex program (8.0.1) satisfying the assumptions from the description of the Ellipsoid algorithm 8.2.2 be solved by the algorithm. Let*

$$N(\epsilon) = \lfloor 2n(n-1)\ln\left(\frac{\mathcal{V}}{\epsilon}\right)\rfloor,$$

where $0 < \epsilon < 1$ *and*

$$\mathcal{V} = \left[\frac{\mathrm{Vol}(E(c_0, B_0))}{\mathrm{Vol}(\hat{G})}\right]^{1/n}$$

Then, for any $\epsilon \in (0, 1)$*, the approximate solution* $x^{N(\epsilon)}$ *found by the method in course of the first* $N(\epsilon)$ *steps, is well-defined and is an* ϵ*-solution to the problem, i.e., belongs to* $\hat{G}$ *and satisfies the inequality*

$$f(x^{N(\epsilon)}) - \min_{\hat{G}} f \leq \epsilon[\max_{\hat{G}} f - \min_{\hat{G}} f]. \tag{8.2.15}$$

8.3 Ellipsoid method and Complexity of Convex Programming

The Ellipsoid method implies fundamental theoretical results on *complexity* of Convex Programming; let us briefly discuss these theoretical issues.

8.3.1 Complexity: what is it?

When speaking about complexity of a class of computational problems, we are interested to answer the following crucial question:

Given a family $\mathcal{P}$ of problem instances and required accuracy ϵ, what can be said about the computational effort sufficient to solve each instance to the prescribed accuracy?

This is an informal question, of course, and to answer it, we should first formalize the question itself, namely, to say

- What does it mean "a family of problems" and how we measure accuracy of approximate solutions;
- What does it mean "effort of computational process"? What is the model of the process and how the effort is measured?

To realize that these indeed are points which should be clarified, consider the following "universal solution method" (which in fact is the main method to solve problems): think! Is it a "computational method"? What is the "effort" for this method?

There are basically two approaches to formalize the complexity-related notions. The first, which is more convenient for "continuous" computational problems, is as follows (according to our goals, I shall speak only about optimization problems in the form (8.2.1)).

Real Arithmetic Complexity Model:

- A family of problems $\mathcal{P}$ is a set of problems (8.2.1) such that a particular member $p = (f, G)$ of the family is encoded by a finite-dimensional *data vector* $d(p)$. The dimension of the data vector is called the *size* of the instance.

  Example 1. Linear Programming over Reals. Here the instances are of the form

  $$\min_x \left\{ f(x) = c^T x : x \in G = \{x \in \mathbf{R}^n \mid Ax \leq b\} \right\},$$

  A being $m \times n$ matrix and b being m-dimensional vector. The data vector of a problem instance is comprised of the pair n, m and of $n + m + nm$ entries of c, b, A written down in a once for ever fixed order (e.g., first the n entries of c, then the m entries of b and finally the nm entries of A in the row-wise order).

  Example 2. Quadratically Constrained Quadratic Programming. Here the instances are of the form

  $$\min_x f_0(x) = \frac{1}{2} x^T H_0 x - b_0^T x$$

  subject to

  $$x \in G = \{x \in \mathbf{R}^n \mid f_i(x) = \frac{1}{2} x^T H_i x - b_i^T x + c_i \leq 0,\ i = 1, ..., m\},$$

  and the data vector is comprised of n, m and the entries of the matrices H_i, the vectors b_i and the reals c_i written down in a once for ever fixed order.

  The number of examples can be easily extended; basically, the typical families of problems in question are comprised of problems of a "fixed generic analytical structure", and the data vector is the set of numerical coefficients of the (fixed by the description of the family) analytical expressions corresponding to the particular instance in question.

  In what follows we restrict ourselves with the families comprised of *solvable* problems with *bounded* feasible sets.

- An ϵ-solution x to a problem instance $p = (f, G)$ is a feasible ($x \in G$) solution such that

$$f(x) - \min_G f \leq \epsilon[\max_G f - \min_G f];$$

there are other definitions of an ϵ-solution, but let us restrict ourselves with the indicated one.

- A computational algorithm for $\mathcal{P}$ is a program for an idealized computer capable to perform operations of exact real arithmetic (four arithmetic operations and computation of elementary functions like $\sqrt{\cdot}$, $\sin(\cdot)$, $\log(\cdot)$, etc.). When solving a problem instance from $\mathcal{P}$, the algorithm gets on input the data vector of the instance and the required accuracy ϵ and starts to process these data; after finitely many elementary steps the algorithm should terminate and return the vector $x_{\rm out}$ of the corresponding dimension, which should be an ϵ-solution to the input instance. The computational effort of solving the instance is, by definition, the total # of elementary steps (arithmetic operations) performed in course of processing the instance.

The second way to formalize the complexity-related notions, the way which is the most widely used in theoretical Computer Science and in Combinatorial Optimization, is given by

Algorithmic Complexity Model:

- A family of problems is a set of problems (8.2.1) such that a particular member $p = (f, G)$ of the family is encoded by an *integer* $i(p)$; the # L of digits in this integer is the bit size of the instance p.

Example 3. Integer programming. Here the problem instances are of the form

$$\min_x \left\{ f(x) = c^T x : x \in G = \{x \in \mathbf{Z}^n \mid Ax \leq b\} \right\},$$

where $\mathbf{Z}^n$ is the space of n-dimensional vectors with integer entries, c and b are vectors of dimensions n and m with integer entries, and A is an $m \times n$ matrix with integer entries.

To encode the data (which form a collection of integers) as a single integer, one can act as follows:

first, write down the data as a finite sequence of binary integers (similarly to Example 1, where the data were real);

second, encode the resulting *sequence* of binary integers as a *single* integer, e.g., by representing

- binary digit 0 as 00

- binary digit 1 as 11

- blank between integers as 01

- sign – at an integer as 10

Example 4. Linear Programming over Rationals. This is a subfamily of the Linear Programming family (Example 1), where we impose on all the entries of c, b, A the requirement to be rational numbers. To encode a problem instance by a single binary integer, it suffices to write the data as a sequence of binary integers, same as in Example 3 (with the only difference that now every rational element of the data is represented by two sequential integers, its numerator and denominator), and then encode the sequence, as it was explained above, by a single binary integer.

In what follows, speaking about Algorithmic Complexity model, we always assume that

- the families in question are comprised of solvable problems with bounded feasible sets

- any problem instance from the family admits a solution which can be naturally encoded by a binary integer

The second assumption is clearly valid for the Integer Programming (Example 3). It is also valid for Linear Programming over Rationals (Example 4), since it can be easily proved that a solvable LP program with integer data admits a solution with rational entries; and we already know that any finite sequence of integers/rationals can be naturally encoded by a single integer.

- Normally, in Algorithmic Complexity model people are not interested in approximate solutions and speak only about exact ones; consequently, no problem with measuring accuracy occurs.
- A computational algorithm is an algorithm in any of (the equivalent to each other) definitions given in Mathematical Logic; you loose nothing when thinking of a program for an idealized computer which is capable to store in memory as many finite binary words as you need and to perform bit-wise operations with these words. The algorithm as applied to a problem instance p from $\mathcal{P}$ gets on input the code $i(p)$ of the instance and starts to process it; after finitely many elementary steps, the algorithm should terminate and return the code of an exact solution to the instance. The computational effort is measured as the total # of elementary bit-wise operations performed in course of the solution process.

Let me stress the difference between the notions of the size of an instance in the Real Arithmetic and the Algorithmic Complexity models. In the first model, the size of an instance is, basically, the # of real coefficients in the natural analytical representation of the instance, and the size is independent of the numerical values of these coefficients – they may be arbitrary reals. For example the Real Arithmetic size of any LP program over Reals (Example 1) with $n = 2$ variables and $m = 3$ inequality constraints is the dimension of the vector

$$(2, 3, c_1, c_2, b_1, b_2, b_3, A_{11}, A_{12}, A_{21}, A_{22}, A_{31}, A_{32}),$$

i.e., 13. In contrast to this, the Algorithmic Complexity size of an LP program over Rationals (Example 4) with $n = 2$ variables and $m = 3$ constraints can be arbitrarily large, if the rational data are "long".

We can observe similar difference between the measures of computational effort in the Real Arithmetic and the Algorithmic Complexity models. In the first of them, the effort required, say, to add two reals is one, independently of what the reals are. In the second model, we in principle cannot deal with reals – only with integers; and the effort to add two N-digit integers is not 1 – it is proportional to N.

8.3.2 Computational Tractability = Polynomial Solvability

After the main complexity-related notions – family of problems, ϵ-solution, solution algorithm and its computational effort – are formalized, we may ask ourselves

What is the complexity of a given family of problems, i.e., the best possible, over all solution algorithms, computational effort sufficient to solve all instances of a given size to a given accuracy?

Before trying to answer this question, we may ask ourselves a more rough (and, in a sense, more important) question:

Is the family in question computationally tractable, i.e. is its complexity a "reasonable" function of the size of instances?

If the answer to this second question is negative, then we may forget about the first one: if we know that to solve an instance of size l from a family $\mathcal{P}$ it may require *at least* 2^l elementary steps, then we may not bother much whether the actual complexity is 2^l, 2^{5l} or $2^{(2^l)}$: in the first case, the maximal size of instances we are able to solve for sure in reasonable time is something like 30, in the second – 6, in the third – 5; this difference normally means nothing, since the sizes of actually interesting problems usually are hundreds, thousands and tens of thousands.

Now, the notion of "computationally tractable" complexity in theoretical studies of the last 30 years is unanimously understood as the one of "polynomial time" solvability. This latter notion first was introduced in the Algorithmic Complexity model and was defined as follows:

A solution algorithm $\mathcal{A}$ for a family $\mathcal{P}$ of problem instances is called polynomial, if the computational effort required by $\mathcal{A}$ to solve a problem instance of an arbitrary bit size L never exceeds $q(L)$, q being certain polynomial; here all complexity-related notions are understood according to the Algorithmic Complexity model.
$\mathcal{P}$ is called polynomially solvable, if it admits polynomial solution algorithm.

The analogy of this definition for the Real Arithmetic Complexity model is as follows:

A solution algorithms $\mathcal{A}$ for a family $\mathcal{P}$ of problem instances is called R-polynomial, if the computational effort required by $\mathcal{A}$ to solve a problem instance p of an arbitrary size l to an arbitrary accuracy $\epsilon \in (0,1)$ never exceeds

$$q(l)\ln\left(\frac{\mathcal{V}(p)}{\epsilon}\right), \tag{8.3.1}$$

where q is certain polynomial and $\mathcal{V}(p)$ is certain data-dependent quantity; here all complexity-related notions are understood according to the Real Arithmetic Complexity model.

The definition of an R-polynomial algorithm admits a very natural interpretation. The quantity $\ln(\mathcal{V}(p)/\epsilon)$ can be viewed as # of accuracy digits in an ϵ-solution; with this interpretation, (8.3.1) means that *the arithmetic cost per accuracy digit is bounded from above by a polynomial of the dimension of the data vector.*

8.3.3 R-Polynomial Solvability of Convex Programming

Invention of the Ellipsoid method (1976) allowed to establish the following important

Theorem 8.3.1 [R-polynomial solvability of Convex Programming]
Consider a family $\mathcal{P}$ of convex problems (8.2.1) and assume that

(i) *all problem instances $p = (f, G)$ from the family are bounded (i.e., their feasible domains G are bounded) and strictly feasible (i.e., their feasible domains G possess nonempty interiors); moreover, given the data vector $d(p)$ of the instance, one can in polynomial in the size $l_p = \dim d(p)$ of the instance number of arithmetic operations compute an ellipsoid which covers G along with a lower bound $v_p > 0$ for the volume of G;*

(ii) *given an instance $p = (f, G) \in \mathcal{P}$ and a point $x \in \mathbf{R}^{n_p}$, n_p being the # of variables in p, on can in polynomial in l_p number of arithmetic operations*

- *imitate the Separation oracle for G on the input x, i.e., check the inclusion $x \in G$ and, if it is not valid, compute a separator e:*
$$\sup_{y \in G} e^T y < e^T x;$$
- *imitate the First order oracle for f, i.e., compute $f(x)$ and $f'(x)$.*

Then there exists an R-polynomial algorithm for $\mathcal{P}$.

Proof is immediate: it suffices to use the Ellipsoid algorithm. Indeed,

- (i) says that, given on input the data vector $d(p)$ of a problem instance $p \in \mathcal{P}$, we can in polynomial in $l_p = \dim d(p)$ number of arithmetic operations compute an ellipsoid $E(c_0, B_0)$ and thus start the Ellipsoid algorithm. As a byproduct, we observe that the number of variables n_p in p is bounded from above by a polynomial of the size l_p of the instance (otherwise it would require too many operations simply to write down the center of the ellipsoid).
- according to the description of the method, to perform a step of it, we should imitate the Separation and, possibly, the First order oracle at current point x_t (according to (iii), it takes polynomial in l_p number of arithmetic operations) and then should perform $O(n_p^2)$ arithmetic operations to update the previous ellipsoid into the current one according to Algorithm 8.2.1.3); by virtue of the previous remark, n_p is bounded from above by a polynomial in l_p, so that the overall arithmetic effort at a step is polynomial in l_p;
- according to Theorem 8.2.1, the method will find an ϵ-solution to p in the number of steps not exceeding
$$N(\epsilon) = \lfloor 2(n_p - 1)n_p \ln\left(\frac{\mathcal{V}}{\epsilon}\right) \rfloor, \quad \mathcal{V} = \left[\frac{\mathrm{Vol}(E(c_0, B_0))}{\mathrm{Vol}(G)}\right]^{1/n_p};$$

we have

$$N(\epsilon) \le N'(\epsilon) = \lfloor 2(n_p - 1)n_p \ln\left(\frac{\mathrm{Vol}^{1/n_p}(E(c_0, B_0))}{v_p^{1/n_p}\epsilon}\right) \rfloor$$

(see (i)); according to (i), we can compute $N'(\epsilon)$ in polynomial in l_p number of operations and terminate the process after $N'(p)$ steps, thus getting an ϵ-solution to p.

The overall arithmetic effort to find an ϵ-solution to p, in view of the above remarks, is bounded from above by

$$r(l_p)N'(\epsilon) \le q(l_p)\ln\left(\frac{\mathcal{V}(p)}{\epsilon}\right), \quad \mathcal{V}(p) \equiv \left[\frac{\mathrm{Vol}(E(c_0, B_0))}{v_p}\right]^{1/n_p},$$

both $r(\cdot)$ and $q(\cdot)$ being certain polynomials, so that the presented method indeed is R-polynomial. ■

When thinking of Theorem 8.3.1, you should take into account that the "unpleasant" assumptions (i) are completely technical and normally can be ensured by slight regularization of problem instances. Assumption (ii) is satisfied in all "non-pathological" applications, so that in fact Theorem 8.3.1 can be qualified as a General Theorem on R-Polynomial Solvability of Convex Programming. I should add that this is, in a sense, an "existence theorem": it claims that in Convex Programming there exists a "universal" R-polynomial solution routine, but it does not say that this is the best possible R-polynomial routine for all particular cases. The main practical drawback of the Ellipsoid method is that it is "slow" – the # of iterations required to solve the problem within a once for ever fixed accuracy ϵ grows quadratically with the dimension n of the problem. This quadratic in n growth of the effort makes it basically impossible to use the method as a practical tool in dimensions like hundreds and thousands. Recently, for many important families of Convex Programming problems (Linear, Quadratically Constrained Quadratic, Geometric and some others) more specialized and more efficient *interior point polynomial algorithms* were developed; these methods are capable to struggle with large-scale problems of real-world origin.

I would say that the Ellipsoid method, as a practical tool, is fine for not large – up to 20-30 variables – convex problems. The advantages of the method are:

- simplicity in implementation and good numerical stability
- universality
- low order of dependence of computational effort on m – the # of functional constraints

  The latter remark relates to Algorithm 8.2.2): the number of steps required to achieve a given accuracy is simply independent of m, and the effort per step is at most proportional to the m (the only place where the number of constraints influence the effort is the necessity to check feasibility of the current iterate and to point out a violated constraint, if any exists). As a result, for "high and narrow" convex programs (say, with up to 20-30 variables and as many thousands of constraints as you wish) the Ellipsoid method seems to be one of the best known Convex Programming routines.

8.4 Polynomial solvability of Linear Programming

Surprisingly, the most important "complexity consequences" of the Ellipsoid method – which is a purely "continuous optimization" algorithm – relate to *Algorithmic Complexity*. The most known of these consequences is

8.4.1 Polynomial Solvability of Linear Programming over Rationals

8.4.1.1 Some History

Everybody knows that Linear Programming programs – both with real and rational data – can be solved efficiently (and are solved almost for 50 years) by the Simplex Algorithm, which is a finite Real Arithmetic routine for LP capable to solve huge practical linear programs. What I am not sure is that everybody

knows what does it mean "efficiently" in the above sentence. This is *practical efficiency* – when solving an LP program

$$\min_x \left\{c^T x : Ax \leq b\right\} \tag{8.4.1}$$

with n variables and $m > n$ inequality constraints, the Simplex *normally* performs $2m$ - $4m$ iterations to find the solution. Anyhow, theoretically Simplex is not a polynomial algorithm: since the beginning of sixties, it is known that there exist simple LP programs p_n with $n = 1, 2, 3, ...$ variables and integer data of the total bit size $L_n = O(n^2)$ such that some versions of the Simplex method solve p_n in no less than 2^n steps. Consequently, the aforementioned versions of the Simplex method are not polynomial – for a polynomial algorithm, the solution time should be bounded by a polynomial of $L_n = O(n)$, i.e., of n. Similar "bad examples" were constructed for other versions of the Simplex method; and although nobody was able to prove that *no* version of Simplex can be polynomial, nobody was also lucky to point out a polynomial version of the Simplex method. Moreover, since mid-sixties, where the Algorithmic Complexity approach became standard, and till 1979 nobody knew whether LP over Rationals (Example 4 above) is or is not polynomially solvable.

8.4.2 Khachiyan's Theorem

The positive answer on the fundamental question whether LP over Rationals is polynomially solvable was given only in 1979 by L. Khachiyan; the main tool used in the proof was the Ellipsoid method invented 3 years earlier.

Below I sketch the reasoning of Khachiyan.

8.4.2.1 Step 1: from Optimization to Feasibility

The first step of the construction reduces the *optimization LP program* (8.4.1) to the *Feasibility program* for a system of linear inequalities. To this end it suffices to write down both the inequalities of (8.4.1) and of its LP dual and to replace minimization in the primal problem and maximization in the dual one with the linear constraint that the duality gap should be zero. This results in the following system of linear inequalities and equations:

$$Ax \leq b; \quad A^T y = c; \quad y \leq 0; \quad c^T x = b^T y \tag{8.4.2}$$

with unknowns $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$. The Duality Theorem in Linear Programming says that the original problem (8.4.1) is solvable if and only if (8.4.2) is, and if it is the case, then x-components of the solutions to (8.4.2) are exactly the optimal solutions to (8.4.1).

Assembling x and y in a single vector of variables u, representing u as the difference of new *nonnegative* variable vectors v and w and, finally, assembling v and w into single vector z (all these manipulations are standard tricks in LP),

we can rewrite (8.4.2) equivalently as the problem

$$\text{find} \;\; z \in \mathbf{R}^N : \;\; Qz \leq q, \;\; z \geq 0; \tag{8.4.3}$$

here $N = 2(n + m)$, and Q, q are certain matrix and vector with rational entries coming, in an evident fashion, from the initial data c, b, A. Note that the bit size L_1 of the data in (8.4.3) is of order of the bit size L of the initial data in (8.4.1); in fact $L_1 \leq 10L$.

It is convenient to pass from problem (8.4.3) with rational data to an equivalent problem with integer data; to this end it suffices to compute the common denominator of all the fractions involved into (8.4.3) and to multiply all the data by this common denominator. As a result, we come to the problem

$$\text{find} \;\; x \in \mathbf{R}^N : \;\; Rz \leq r, \;\; z \geq 0; \tag{8.4.4}$$

the properties of this problem are as follows:

- (8.4.3) is equivalent to the LP program (8.4.1): both the problems are solvable/unsolvable simultaneously; if they are solvable, then any solution to (8.4.4) can be converted in an explicit and simple manner into optimal solution to (8.4.1);

- the data in (8.4.4) are integer, and their bit size L_2 is bounded by a polynomial (something like $200L^2$) of the bit size L of the initial data [4].

It follows that *if we were able to solve in polynomial time the Feasibility problem (8.4.4), we would get, as a byproduct, a polynomial algorithm for the initial problem (8.4.1).*

8.4.2.2 Step 2: from Feasibility to Solvability

Instead of attacking the Feasibility problem (8.4.4) (where we are asked to check whether (8.4.4) is solvable and if it is the case – are asked to point out a solution), let us for the time being restrict our goals with the following relaxed solvability version of (8.4.4):

(S) *given the data of (8.4.4), detect whether the problem is solvable.*

The simplification, as compared to (8.4.4), is that now we are not obliged to point out an explicit solution to a solvable instance.

Problem (S) is the problem which is solved in polynomial time by the Ellipsoid method. This is done as follows.

- **From (S) to (S′).** Problem (S) clearly is equivalent to the following optimization problem: let

$$f(z) = \max\left[(Rz)_1 - r_1|; (Rz)_2 - r_2; ...; (Rz)_M - r_M|\right]$$

(M is the number of rows in R) be the residual function of the system of inequalities $Rz \le r$; this function clearly is nonnegative at a nonnegative z if and only if z is a feasible solution to (8.4.4). Consequently, (S) is exactly the following problem:

(S′): detect whether $\min\limits_{z\in\mathbf{R}^N, z\ge 0} f(z) \le 0$

((S) is solvable if and only if the answer in (S′) is "yes").

- **From (S′) to (S″).** The next step is given by the following simple

Lemma 8.4.1 *The answer in (S′) is "yes" if and only if it is "yes" in the following problem*

(S″) detect whether $\min\limits_{z:0\le z_i\le 2^{L_2},\, i=1,...,N} f(z) \le 0$

L_2 being the bit length of the data in (S).

Proof. If the answer in (S″) is "yes", then, of course, the answer in (S′) also is "yes". It remains to prove, therefore, that if the answer in (S′) is "yes" (or, which is the same, if (S) is solvable), then the answer in (S″) also is "yes". This is easy. The solution set of (S), being nonempty, is a nonempty closed convex polyhedral set (here and in what follows I use the standard terminology of Convex Analysis; this terminology, along with all required facts, was presented in the course Optimization I); since (S) involves nonnegativity constraints, this set does not contain lines, and therefore, due to the well-known fact of Convex Analysis, possesses an extreme point $\bar{z}$. From the standard facts on the structure of extreme points of a polyhedral set given by (8.4.4) it is known that the vector z^* comprised of nonzero entries of $\bar{z}$, if such entries exist, satisfy *nonsingular* system of linear equations of the form

$$\bar{R}z^* = \bar{r},$$

where

[4] the latter property comes from the fact that the common denominator of the entries in (8.4.3) is an integer of bit size at most $L_1 \le 10L$; therefore when passing from (8.4.3) to (8.4.4), we increase the bit size of each entry by at most $10L$. Since even the total bit size of the entries in (8.4.3) is at most $10L$, the bit size of an entry in (8.4.4) is at most $20L$; and there are at most $10L$ entries. All our estimates are extremely rough, but it does not matter – all we are interested in is to ensure polynomiality of the bit size of the transformed problem in the bit size of the initial one

– $\bar{R}$ is a $k \times k$ nonsingular submatrix in R (k is the # of nonzero entries in $\bar{z}$)

– $\bar{r}$ is certain fragment of the right hand side vector r

(R and r are given by (8.4.4)).

It follows that the entries in z^* are given by the Cramer rules – they are ratios of certain $k \times k$ determinants taken from the $k \times (k+1)$ matrix $[\bar{R}|\bar{r}]$:

$$z_i^* = \frac{\Delta_i}{\Delta_0}.$$

All entries in this matrix are integers, and the total bit size of the entries does not exceed L_2. It follows that all the determinants are, in absolute value, at most 2^{L_2} [5]. Thus, the numerators in the Cramer formulae are $\le L_2$ in absolute values, while the denominator (being a nonzero integer) is in absolute value ≥ 1. Consequently, $|z_i^*| \le 2^{L_2}$.

Thus, all nonzero entries in $\bar{z}$ are $\le 2^{L_2}$ in absolute values. Since $\bar{z}$ is a solution to (S), this is a point where f is nonnegative. We conclude that if the answer in (S) is "yes", then f attains nonpositive value in the box $0 \le z_i \le 2^{L_2}$, $1 \le i \le N$, so that the answer in (S$''$) also is "yes". ∎

- **(S$''$) as a convex program.** To solve problem (S$''$) is exactly the same as to check whether the optimal value in the optimization program

$$\min_x \left\{ f(z) : z \in G = \{z \in \mathbf{R}^n \mid 0 \le z_i \le 2^{L_2},\ i = 1, ..., N\} \right\} \tag{8.4.5}$$

is or is not positive. The objective in the problem is easily computable convex function (since it is maximum of M linear forms), and the domain G of the problem is a simple solid - a box. Remark 8.1.1 explains how to imitate the First order and the Separation oracles for the problem; we can immediately point out the initial ellipsoid which contains G (simply the Euclidean ball circumscribed around the cube G). Thus, we are able to solve the problem by the Ellipsoid method. From Theorem 8.2.1 (where one should estimate the quantities $\mathcal{V}$ and $\max_G f - \min_G f$ via L_2; this is quite straightforward) it follows that in order to approximate the optimal value f^* in (8.4.5) within a prescribed *absolute* accuracy $\nu > 0$, it suffices to perform

$$N_\nu = O(1)N^2[L_2 + \ln \frac{1}{\nu}]$$

steps with at most $O(1)(M+N)N$ arithmetic operations per step, which gives totally

$$\mathcal{M}(\nu) = O(1)N^3(M+N)[L_2 + \ln \frac{1}{\nu}] \tag{8.4.6}$$

arithmetic operations ($O(1)$ here and in what follows are positive absolute constants).

All this looks fine, but in order to detect whether the optimal value f^* in (8.4.5) is or is not non-positive (i.e., whether (S) is or is not solvable), we should distinguish between two "infinitesimally close to each other" hypotheses $f^* \le 0$ and $f^* > 0$, which seems to require the exact value of f^*;

[5] This is a consequence of the Hadamard inequality: the absolute value of a determinant (≡ the volume of the parallelotope spanned by the rows of the determinant) does not exceed the product of the Euclidean lengths of the rows of the determinant (product of the edges of the parallelotope). Consequently, $\log_2|\Delta_i|$ does not exceed the sum of binary logs of the Euclidean lengths of the rows of $[\bar{R}|\bar{r}]$. It remains to note that the binary logarithm of the Euclidean length of an integer vector clearly does not exceed the total bit length of the vector:

$$\frac{1}{2}\log_2(a_1^2 + ... + a_k^2) \le \frac{1}{2}\log_2[(1+a_1^2)(1+a_2)^2...(1+a_k^2)] = \sum_{i=1}^k \frac{1}{2}\log_2[1+a_i^2] \le \sum_{i=1}^k \log_2[1+|a_i|]$$

and the latter expression clearly is $\le$ the total # of binary digits in integers $a_1, ..., a_k$.

and all we can do is to approximate "quickly" f^* to a prescribed accuracy $\nu > 0$, not to find f^* exactly.

Fortunately, our two hypotheses are not infinitesimally close to each other – there is a "gap" between them. Namely, it can be proved that

if the optimal value f^* in (8.4.5) is positive, it is not too small, namely $f^* \geq 2^{-\pi(L_2)}$ with certain polynomial $\pi(\cdot)$ [6].

The latter remark says that to distinguish between the cases $f^* \leq 0$ and $f^* > 0$ means in fact to distinguish between $f^* \leq 0$ and $f^* \geq 2^{-\pi(L_2)}$; and to this end it suffices to restore f^* within absolute accuracy like $\nu = 2^{-\pi(L_2)-2}$. According to (8.4.6), this requires $O(1)N^3(N+M)[L_2 + \pi(L_2)]$ arithmetic operations, which is not more than a polynomial of L_2 and, consequently, of L (since L_2 is bounded from above by a polynomial of L).

Thus, *we indeed can decide whether f^* is or is not nonpositive or, which is the same, whether* (S) *is or is not solvable, in polynomial in L number of arithmetic operations.*

This is not exactly what we need: our complexity model counts not arithmetic, but bit-wise operations. It turns out, anyhow (the verification is straightforward, although very dull), that *one can apply the Ellipsoid method with inexact arithmetic instead of the exact one, rounding the intermediate results to a polynomial in L number of accuracy digits, and it still allows to restore f^* within required accuracy.* With the resulting inexact computations, the bit-wise overall effort becomes polynomial in L, and the method becomes polynomial.

Thus, (S) *is polynomially solvable.*

8.4.2.3 From Solvability back to Feasibility

It remains to explain how, given possibility to solve in polynomial time the Solvability problem (S), one could solve in polynomial time the Feasibility problem (8.4.4) and thus – the original problem (8.4.1).

First of all, we solve (S) – we already know how to do it in polynomial time. If the answer in (S) is "no", then (8.4.4) is unsolvable, and we are done. It remains to consider the case when the answer in (S) is "yes"; here we should point out explicitly the solution to the system

$$(*_0): \qquad Pz \leq p, \ z \geq 0.$$

Let us act as follows. Take the first inequality $P_1^T z \leq p_1$ in the system $Pz \leq p$ and make it equality. This will give us a new system $(*_1^+)$ of, basically, the same bit size. as the one of $(*_0)$. Let us check in polynomial time whether this new system is solvable. If it is so, let $(*_1)$ be the system obtained from $(*_0)$ by replacing the first inequality with the equality; note that this system is solvable, and all solutions to it are solutions to $(*_0)$. If $(*_0^+)$ is unsolvable, it means that the hyperplane $\Pi = \{P_1^T z = p_1\}$ does not intersect the solution set of $(*_0)$. Since the latter set is nonempty and convex, the only possibility for the hyperplane Π not to intersect this set is when the inequality $P_1^T z \leq p_1$ is redundant in system $(*_0)$ – when eliminating this inequality from the system, we do not vary the solution set. If it is the case, let us define $(*_1)$ as the system obtained from $(*_0)$ by eliminating the first inequality.

Let us look what we get. Solving in polynomial time problem (S) associated with the system $(*_1^+)$ with basically the same bit size as the one of (8.4.4), we have updated the initial system of inequalities

[6] The proof (I skip the details) is completely similar to that one of Lemma 8.4.1: we again exploit the fact that f^* is the optimal value in certain LP program with integer data of the polynomial in L_2 total bit size, namely, in the program

$$\min_{t,z} \left\{ t : t \geq (Pz)_i - p_i, \ i = 1, ..., N, \ 0 \leq z_i \leq 2^{L_2} \right\}.$$

The optimal value in this problem is achieved at an extreme point, and this point, same as in Lemma 8.4.1, is rational with not too large numerators and denominators of the entries. Consequently, the optimal value of t is rational with not too large numerator and denominator, and such a fraction, if positive, of course, is not too close to 0.

$(*_0)$ into a new system $(*_1)$; this new system is solvable, and every solution to it is a solution to $(*_0)$ as well, and the new system has one inequality less than the initial system.

Now let us apply the same trick to system $(*_1)$, trying to make the equality the second inequality $P_2^T z \leq p_2$ of the initial system; as a result, we will "kill" this second inequality – either make it the equality, or eliminate it at all – and all solutions to the resulting system $(*_2)$ (which for sure will be solvable) will be solutions to $(*_1)$ and, consequently, to the initial system $(*_0)$.

Proceeding in this manner, we in M steps (M is the row size of P) will "kill" all the inequalities $Pz \leq p$ - some of them will be eliminated, and some - transformed into equalities. Now let us kill in the same fashion the inequalities $z_i \geq 0$, $i = 1, ..., N$. As a result, in N more steps we shall "kill" all inequalities in the original system $(*_0)$, including the nonnegativity ones, and shall end up with a system of *linear equations*. According to our construction, the resulting system $(*_{M+N})$ will be solvable and every solution to it will be a solution to $(*_0)$.

It follows that to get a solution to $(*_0)$ it remains to solve the resulting solvable system $(*_{M+N})$ of linear equations by any standard Linear Algebra routine (all these routines are polynomial).

Note that the overall process requires to solve $N + M$ Solvability problems of, basically, the same bit size as (8.4.4) and to solve a system of linear equations, again of the same bit size as (8.4.4); thus, the overall complexity is polynomial in the size of (8.4.4).

The proof of polynomial time solvability of LP over Rationals – which might look long and dull – in fact uses absolutely simple and standard arguments; the only nonstandard – and the key one – argument is the Ellipsoid method.

8.4.3 More History

Although the Ellipsoid method as applied to LP turns out to be polynomial – and therefore, theoretically, much more efficient than the non-polynomial Simplex method – in practical computations the Simplex (which – it is a kind of mystery why – never works according to its disastrous theoretical worst case efficiency bound) is incredibly better than the Ellipsoid method, so that it makes absolutely no sense to think that the Ellipsoid method can be competitive with Simplex as a practical LP tool. Nevertheless, the "complexity approach" to LP proved itself to be fruitful not only in theory. In 1984, N. Karmarkar developed a new polynomial time algorithm for LP – the first of the interior point methods which are in great fashion now – and this method turned out to be quite competitive with the Simplex method in practical computations, not speaking about great theoretical advantage of polynomiality shared by the method of Karmarkar.

Lecture 9

Active Set and Penalty/Barrier Methods

The traditional methods for general type constrained minimization problems

$$\begin{array}{lrcl} (P) & & & \\ & f(x) & \to & \min \\ \text{s.t.} & & & \\ & h_i(x) & = & 0,\ i = 1, ..., m, \\ & g_j(x) & < & 0,\ j = 1, ..., k \end{array}$$

with $x \in \mathbf{R}^n$ (when saying "general type problems", I mean not necessarily convex ones; Convex Programming is another and much nicer story) can be, roughly speaking, separated into the following four groups:

- primal methods, where we try to act similarly to what we did in the unconstrained case – i.e., to move along the feasible set in a way which ensures, at each step, progress in the objective;
- barrier and penalty methods, where we reduce (P) to a series of "approximating" the problem unconstrained programs;
- Lagrange multipliers methods, where we focus on the *dual problem* associated with (P); this dual problem is either unconstrained one (when (P) is equality constrained), or has simple nonnegativity constraints (when (P) includes inequalities) and is therefore simpler than (P). When solving the dual problem, we get, as a byproduct, approximate solutions to (P) itself. Note that a posteriori the Lagrange multiplier methods, same as the penalty/barrier ones, reduce (P) to a sequence of unconstrained problems, but in a "smart" manner quite different from the straightforward penalty/barrier scheme;
- SQP (Sequential Quadratic Programming) methods. The SQP methods, in contrast to all previous ones, neither try to improve the objective staying within the feasible set, nor approximate the constrained problem by unconstrained ones, but directly solve the KKT system of (nonlinear) equations associated with (P) by a kind of the Newton method.

This lecture is devoted to brief overview of the first two groups of methods; my choice is motivated by the fact that these methods basically directly reduce the constrained problem to a sequence of unconstrained ones, and it is easy to understand them, given our knowledge of technique for unconstrained minimization.

Before passing to the main body of the lecture, let me make an important comment.

> When solving an unconstrained minimization problem, we were aimed to find the optimal solution; but the best we indeed were able to do was to ensure convergence to the set of critical points of the objective, the set of points where the First Order Necessary Optimality

condition – the Fermat rule – is satisfied. Similarly, the best we can do in constrained optimization is *to ensure convergence to the set of KKT points of the problem – those points where the First Order Necessary Optimality condition form Lecture 7 is satisfied.* Whether it fits our actual goal or not – this is another story, sometimes with happy end (e.g., in the case of *convex* problem a KKT point is for sure a globally optimal solution), sometimes – not, but this is all we can achieve.

During all this lecture, we make the following assumption on problem (P) in question:

Regularity: The problem is *regular*, i.e., it is feasible, the objective and the constraints are at least once continuously differentiable, and every feasible solution x is a regular point for the system of constraints: the gradients of the constraints active at x (i.e., satisfied at x as equalities) are linearly independent.

9.1 Primal methods

Just to start our considerations, let me briefly outline the idea of the oldest primal method – the Feasible Directions one.

9.1.1 Methods of Feasible Directions

The Feasible Directions method can be applied only to a problem without *nonlinear* equality constraints. To simplify considerations, assume that all our constraints are *linear inequalities*, so that the problem is

$$\text{(LIP)} \quad \min_x \left\{ f(x) : g_j(x) \equiv a_j^T x - b_j \le 0,\ j = 1, ..., m \right\}. \quad [x \in \mathbf{R}^n] \tag{9.1.1}$$

The idea of the method is as follows: the *First Order Necessary Optimality conditions (the KKT conditions) are "constructive": if they are not satisfied at a given feasible solution x, then we can explicitly point out a better – with a smaller value of the objective – feasible solution x^+.*

Indeed, we remember from the Lecture 7 that the KKT conditions were obtained from the following construction: given feasible solution x, we associate with it *linearization of (LIP) at x* – the auxiliary Linear Programming program

$$(\text{LIP}_x) \quad \min_x \bar{f}(y) \equiv f(x) + (y - x)^T \nabla f(x)$$

subject to

$$g_j(y) \le 0,\ j = 1, ..., m.$$

x is a KKT point of (9.1.1) if and only if x is optimal solution to (P_x) (this was exactly the conjecture which led us to the KKT condition).

From this "if and only if" statement we conclude that if x is not a KKT point of (9.1.1), then x is not optimal solution to (P_x). In other words (pass in (P_x) from variable y to $d = y - x$), there exists a *descent direction* d – direction satisfying

$$d^T \nabla f(x) < 0,\ g_j(x + d) \le 0,\ j = 1, ..., m.$$

When performing a small enough step in this direction, we improve the objective (by the same reasons as in the Gradient Descent) and do not violate the constraints; after such a step is performed, we can iterate the construction at the new point, an so on.

Normally, at a non-KKT point there exist many descent directions. In the Feasible Directions method we choose the one which is "most perspective" – along which the objective decreases at the highest possible rate. Of course, to choose the most perspective direction, we should normalize somehow the candidates. The standard normalization here is given by the restriction

$$|d_i| \le 1,\ i = 1, ..., n, \tag{9.1.2}$$

so that the direction in question is the solution to the following LP program:

$$\min_d \left\{ d^T \nabla f(x) : g_j(x+d) \le 0, j = 1, ..., m, \ |d_i| \le 1, \ i = 1, ..., n \right\}. \tag{9.1.3}$$

Normalization (9.1.2) instead of more natural normalization like $|d| \le 1$ is motivated by the desire for the direction to be determined via an LP, and thus effectively solvable, program.

After the "most perspective" direction d is found, we define the largest stepsize γ, let it be called $\bar{\gamma}$, among the stepsizes which keep the point $x+\gamma d$ feasible, and define the next iterate x^+ via the linesearch applied to f on the segment $[0, \bar{\gamma}]$:

$$x^+ \in \text{Argmin}\{f(y) : y = x + \gamma d, \ 0 \le \gamma \le \bar{\gamma}\}.$$

Then we replace x with x^+ and loop.

The outlined idea can be naturally extended onto inequality constrained problems with nonlinear constraints, but we are not going to discuss this issue in more details. What should be said is that Feasible Directions methods almost never are used in practice – too often they turn out to be too slow, even in "good" – linearly constrained – situations. Much better methods for linearly constrained optimization are *Active Set* methods which we are about to consider.

9.1.2 Active Set Methods

Consider a linearly constrained problem (P) – one with linear inequality and equality constraints, and let G be the feasible domain of the problem. For the sake of convenience, assume that there are no equality constraints at all – we always can make it the case replacing the entire $\mathbf{R}^n$ by the affine set given by the equality constraints.

The feasible domain G of the problem is given by finite set of linear inequalities, so that it is a polyhedral set. We may classify all feasible solutions to the problem – all points of G – according to which constraints are active at these solutions, so that an arbitrary subset I of the set $\mathcal{I} = \{1, ..., m\}$ of constraint indices defines a "face" G_I in G; G_I is comprised of all feasible solutions x such that the constraints g_i with indices from I are active at x (when I is empty, then, by definition, G_I is the entire G). For some I, the faces S_I can be empty; from now on, we shall not be interested in these "empty faces" and will call faces only those of the sets G_I which are nonempty. For each face G_I, let S_I be affine set given by the system of linear equalities

$$g_i(x) = 0, \ i \in I,$$

so that G_I is a polyhedral set in S_I; to get this set, we should add to the linear equalities defining S_I linear inequalities

$$g_i(x) \le 0, \ i \notin I.$$

From the Regularity assumption it follows that a face G_I is a polyhedral set of affine dimension exactly $n - \text{Card}(I)$, where $\text{Card}(I)$ is the number of elements in the set I, and that affine dimension of the affine set S_I also is $n - \text{Card}(I)$; in other words, S_I is the affine span of G_I. $G_\emptyset$ – the only face of affine dimension n - is the entire G; the boundary of G is comprised of *facets* – $(n-1)$-dimensional polyhedral sets G_I associated with one-element sets I. The relative boundary of a facet is comprised of $(n-2)$-dimensional faces G_I corresponding to 2-element sets I, and so on, until we come to 1-dimensional *edges* – faces G_I associated with $(n-2)$-element sets I – and vertices – faces of dimension 0 given by n-element sets I. The simplest way to get an impression of this face structure of G is to imagine a 3D cube given by 6 linear inequalities $x_i - 1 \le 0$, $-1 - x_1 \le 0$, $i = 1, 2, 3$. Here we have one 3-dimensional facet – the entire cube; it is bounded by 6 2-dimensional facets, each of them being bounded by 4 1-dimensional edges (there are totally 12 of them). And edges, in turn, are bounded by 0-dimensional vertices (totally 8).

9.1.2.1 Active Set scheme: the idea

Now let us look at our optimization problem, where we should minimize a smooth objective over G. Just to explain the idea of the active set methods, assume for a moment that the objective is strongly convex,

so that there is exactly one KKT point x^* in our problem, which is the global solution. Let us denote by I^* the set of indices of those constraints which are active at x^*; then x^* belongs to the relative interior of the face G_{I^*} (the constraints active at x^* participate in the description of the affine span S_{I^*} of G_{I^*}, and the remaining constraints are satisfied at x^* as strict inequalities). Since x^* is globally optimal solution to our problem, it is also optimal solution to the problem

$$\min_x \{f(x) : x \in G_{I^*}\},$$

(since the feasible set of the latter problem contains x^* and is contained in G), and since x^* is a relative interior point of G_{I^*}, it is a local solution to the problem

$$(P_{I^*}) \quad \min_x \{f(x) : x \in S_{I^*}\}.$$

But the objective f is strongly convex, so that x^* is a local solution to the latter problem if and only if it is a global solution to it, and such a solution is unique. It follows that *if we were clever enough to guess in advance what is I^*, we could solve instead of our original problem (P) the problem (P_{I^*}).* Now, the problem (P_{I^*}) is actually unconstrained – we could choose orthogonal coordinates in the affine set S_{I^*}, express the n original coordinates of a point $x \in S_{I^*}$ as linear functions of these new coordinates and substitute these expressions into f, thus getting a function $\bar{f}$ of $k = \dim S_{I^*}$ new coordinates – the restriction of f onto S_{I^*}. Problem (P_{I^*}) clearly is equivalent to unconstrained minimization of $\bar{f}$, so that it can be solved via unconstrained minimization machinery already known to us[1)].

We see that if we were clever enough to guess what are the constraints active at the optimal solution, we would be able immediately reduce our constrained problem to an unconstrained one which we already know how to solve. The difficulty, of course, is that we never are that clever to guess the active constraints. The idea of the Active Set methods is *to use at every iteration t certain current guess I_{t-1} of the actual "active set" I^* and act as if we were sure that our guess is exact, until current information will say to us that the guess is false. When it happens, we somehow update the guess and proceed with the new one, and so on.* Now let us look how this idea is implemented.

9.1.2.2 Active Set scheme: implementation

At an iteration t of an Active Set method, we have current *feasible* iterate x_{t-1} along with current *working set* I_{t-1} – a subset of the set $\mathcal{I} = \{1, ..., m\}$ of indices of our constraints; I_{t-1} is comprised exactly of indices of the constraints which are active at x_{t-1}. At the iteration, we perform from x_{t-1} one step of a chosen method for unconstrained minimization as applied to the problem

$$(P_t) \quad \min_x \left\{f(x) : x \in S^{t-1} \equiv S_{I_{t-1}}\right\};$$

as it was already explained, this latter problem is, essentially, an unconstrained one. The only assumption on the method we use is that it is of our standard structure – if x_{t-1} is not a critical point of f on our current *working plane* $S^{t-1} \equiv S_{I_{t-1}}$, i.e., if the orthogonal projection of $\nabla f(x_{t-1})$ on the working plane is nonzero, the method chooses somehow a descent direction d_t of f at x_{t-1} in the working plane and uses a kind of line search to perform a step in this direction which reduces the value of the objective[2)].

Assume that x_{t-1} is <u>not</u> a critical point of f on the current working plane S^{t-1}, so that d_t is well-defined descent direction of the objective. When performing line search in this direction, let us impose

---

[1)] In actual computations, people do not use literally this way to solve (P_{I^*}) – instead of explicit parameterization of S_{I^*} and then running, say, a Quasi-Newton method in the new coordinates, it is computationally better to work with the original coordinates, modifying the method accordingly; this algorithmic difference has no importance for us, since the trajectory we get is exactly the same trajectory we would get when parameterizing S_{I^*} and running the usual Quasi-Newton in the coordinates parameterizing the affine set S_{I^*}.

[2)] Please note slight incorrectness in the above description. Rigorously speaking, we were supposed to say "orthogonal projection of the gradient on the linear subspace of directions parallel to the working plane" instead of "orthogonal projection of the gradient on the working plane" and "direction d_t parallel to the working plane" instead of "direction d_t in the working plane"; exact formulations would be too long

on the stepsize γ_t the restriction as follows:

$$\gamma_t \leq \gamma_t^*,$$

where γ_t^* is the *largest* stepsize which keeps the shifted point in the face $G^{t-1} \equiv G_{I_{t-1}}$:

$$\gamma_t^* = \sup\{\gamma \geq 0 \mid g_i(x_{t-1} + \gamma d_t) \leq 0,\ i \notin I_{t-1}\}.$$

Note that the situation is as follows: since I_{t-1} is exactly the set of indices of the constraints active at x_{t-1}, x_{t-1} is relative interior point of the face G^{t-1}; since the direction d_t is a direction in the current working plane S^{t-1}, an *arbitrary* step from x_{t-1} in this direction keeps the shifted point in S^{t-1} – i.e., the shifted point for sure satisfies the constraints which were active at x_{t-1}. A too large step, however, may make the shifted point to violate one or more of the constraints which were *non*active at x^*; γ_t^* is exactly the largest of those steps for which this bad thing does not occur. It is clear that small enough steps from x_{t-1} in the direction d_t keep the shifted point within the feasible domain, so that γ_t^* is positive. It may happen that γ_t^* is $+\infty$ – an arbitrarily large step in the direction d_t keeps the shifted point feasible; but in the case when γ_t^* is finite, the point $x_t^+ = x_{t-1} + \gamma_t^* d_t$ for sure belongs to the relative boundary of the face G^{t-1}, i.e., the set of constraints active at x_t^+ is strictly larger than the set $\{g_i, i \in I_{t-1}\}$ of constraints active at x_{t-1}.

Now let us look at the new iterate

$$x_t = x_{t-1} + \gamma_t d_t$$

given by our method for unconstrained minimization with the above restriction on the stepsize. There are two possibilities:

- $\gamma_t < \gamma_t^*$. It means that x_t is in the relative interior of the face G^{t-1}, so that the set of indices I_t of the constraints active at x_t is exactly the same as I_{t-1}, and the new working plane $S^t = S_{I_t}$ is the same as the old working plane S^{t-1}; in this case the next iteration will deal with the same problem (P_t) and will simply perform a new step of the chosen method for unconstrained minimization of f over the old working plane;

- $\gamma_t = \gamma_t^*$. It means that x_t is on the relative boundary of the face G^{t-1}, i.e., that the set I_t of indices of the constraints active at x_t is strictly larger than I_{t-1} – some constraints which were nonactive at x_{t-1} become active at x_t. In this case the new working plane $S^t = S_{I_t}$ is strictly less than the previous one; in other words, we have *shrunk* the working plane, and the next iteration will deal with a new problem (P_{t+1}) – the one of minimizing f on the new working plane. In other words, *we have corrected our guess for the actual active set I^* by extending the previous guess I_{t-1} to a new guess I_t.*

Now let us look what will eventually happen in the outlined process. During a number of iterations, it will work with our initial guess I_0, solving the problem of unconstrained minimization of f on the initial working plane S_{I_0}. Then the guess will be corrected, the working plane – shrunk, and the method will switch to minimization of f on this new working plane. After several steps more, the working plane again will be shrunk, and so on. It is clear that there could be at most n switches of this type – the initial working plane is of the dimension at most n, and each time it is shrunk, the dimension of the new working plane is reduced at least by 1. It follows that, starting from some iteration, the method all the time will work with a fixed working plane and will solve the problem of minimizing f on this working plane. Assuming the method for unconstrained minimization we use globally converging, we conclude that the trajectory of the method

- either will be finite and will terminate at a critical point of f at the current working plane,

- or will be infinite, and all limiting points of the trajectory will be critical points of the objective at the current working plane.

With a reasonable idealization, we may assume that in course of running the method, we eventually will

find a critical point of the objective at the current working plane[3)]. Let us look what can be the situation when it happens – our current iterate x_t turns out to be a critical point of f on the current working plane S^t. In other words, assume that the gradient of f at the point x_t is orthogonal to all directions parallel to the affine plane

$$S^t = \{x \mid g_i(x) = 0,\ i \in I_t\},$$

I_t being the set of indices of the constraints active at x_t. The set of directions parallel to S^t is the linear subspace

$$L^t = \{d \mid d^T \nabla g_i = 0,\ i \in I_t\},$$

and the vector $\nabla f(x_t)$ is orthogonal to L^t if and only if it is a linear combination of the vectors ∇g_i, $i \in I_t$ (see the results of Lecture 1 on the structure of the orthogonal complement to a linear subspace given by a system of homogeneous linear equations). In other words, we have

$$\nabla f(x_t) + \sum_{i \in I_t} \lambda_i \nabla g_i = 0 \tag{9.1.4}$$

with properly chosen coefficients λ_i. Note that these coefficients are uniquely defined, since from Regularity assumption it follows that the vectors ∇g_i, $i \in I_t$, are linearly independent.

There are two possibilities:

- A. The coefficients λ_i, $i \in I_t$, are nonnegative. It means that x_t is a KKT of the problem (P), λ_i being the Lagrange multipliers certifying this fact.
- B. At least one of the coefficients λ_i is negative.

In the case A we have reached our target – have found a KKT point of the original problem – and can terminate. What to do in the case B? The answer is as follows: *in this case we can easily find a new iterate which is feasible for (P) and is better than x_t from the viewpoint of the objective*. To see it, assume for the sake of definiteness that $1 \in I_t$ and λ_1 is negative, and let d be the orthogonal projection of $-\nabla f(x_t)$ on the plane

$$L_I = \{h \mid h^T \nabla g_i = 0,\ i \in I\},\ I = I_t \backslash \{1\}.$$

We claim that d is a nonzero direction which is descent for the objective and is such that

$$d^T \nabla g_1 < 0. \tag{9.1.5}$$

Before proving this simple fact, let us look at it consequences. Let us perform a small step $\gamma > 0$ from x_t along the direction d and look what can be said about the shifted point

$$x^+ = x_t + td.$$

Since d belongs to the plane L_I, d is orthogonal to the gradients of the constraints g_i with the indices $i \in I$; all these constraints are active at x_t, and due to the indicated orthogonality they, independently of γ, will be satisfied as active constraints also at x^+. The constraints which were not active at x_t, will remain nonactive also at x^+, provided that γ is small. All we need is to understand what will happen with the constraint g_1 which was active at x_t. This is clear – from (9.1.5) it follows that this constraint will be satisfied and will be nonactive at x^+ whenever $\gamma > 0$. Thus, for small positive γ x^+ will be feasible, and the constraints active at x^+ are exactly those with indices from I. And what about the objective? It also is clear: it was claimed that d is a descent direction of the objective at x_t, so that small enough step from x_t along this direction reduces the objective. We see that the step

$$x_t \mapsto x_{t+1} \equiv x_t + \gamma d$$

[3)] As we shall see, sometimes this is even not an idealization. In the general case it, of course, *is* an idealization: minimizing the objective over a fixed working plane, we normally never will reach the set of *exactly* critical point of f on the plane. In actual computations, we should speak about "ϵ–critical points" – those where the gradient of the objective reduced at the working plane, i.e., the orthogonal projection of ∇f onto the plane, is of norm at most ϵ, ϵ being a small tolerance; such a point indeed will be eventually reached

with properly chosen stepsize $\gamma > 0$ strictly reduces the objective and results in a new iterate x_t, still feasible for (P), with $I_{t+1} = I$. Thus, *when the above process "gets stuck" – reaches a feasible solution which is a critical point of f on the current working plane, but is not a KKT point of (P) – we can "release" the process by a step which improves the objective value and extends the current working plane* (since $I_{t+1} = I$ is strictly less than I_{t-1}, the new working plane S^{t+1} is strictly larger than S^t).

Incorporating in our scheme this mechanism of extending working planes whenever the method reaches a critical point of the objective in the current working plane, but this point is not a KKT point of the problem – we get a descent method (one which travels along the feasible set, each iteration improving the objective) which can terminate only at a KKT point of the problem, and this is what is called an Active Set method. In fact the presented scheme is an "idealized" method – we have assumed that the method for unconstrained minimization we use, being applied to the problem of minimizing f on a fixed working plane, will eventually find a critical point of f on this plane. For a general-type objective this is not the case, and here in actual computations we should speak about "nearly critical, within a given tolerance, points", pass to approximate versions of (9.1.4), etc.

To proceed, we should prove the announced statement about the direction d. This is immediate:

since d is orthogonal projection of $-\nabla f(x_t)$ on L_I, this vector is orthogonal to all ∇g_i with $i \in I$, and the inner product of d with $\nabla f(x_t)$ is $-|d|^2$. With these observations, multiplying (9.1.4) by d, we get

$$-|d|^2 + \lambda_1 d^T \nabla g_1 = 0,$$

and since λ_1 is negative, we conclude that $d^T \nabla g_1$ is negative whenever d is nonzero. To prove that d indeed is nonzero, note that otherwise $\nabla f(x_t)$ would be orthogonal to $L_I = \{\nabla g_i \, i \in I\}^\perp$, i.e., would be a linear combination of the vectors ∇g_i, $i \in I$, or, which is the same, along with equality (9.1.4) there would exist similar equality not involving ∇g_1. This is impossible, since from the Regularity assumption it follows that the coefficients in a relation of the type (9.1.4) are uniquely defined. Thus, (9.1.5) indeed takes place and d is nonzero; since, as we already know, $d^T \nabla f(x_t) = -|d|^2$, d is a descent direction of the objective. ∎

9.1.2.3 Active Set Scheme: convergence

The convergence properties of an Active Set method are given by the following

Theorem 9.1.1 *Assume that linearly constrained problem (P) in question possesses the following property: for every subset I of the index set $\mathcal{I} = \{1, ..., m\}$, the number of those critical points of f on S_I which belong to G_I is finite. Then the idealized Active Set method terminates with a KKT point of (P) in finitely many steps.*

Proof is immediate: the only possibility for the method to terminate is to find a KKT point of the problem. Assume that the method does not terminate, and let us lead this assumption to a contradiction. Let us call a *phase* the sequence of iterates between two subsequent extensions of the working plane. If the method does not terminate, the number of phases clearly is infinite: indeed, during a phase, the method for some (according to our idealization, finite) number of iterations deals with certain working plane, then shrinks it and deals for finitely many iterations with the shrunk working plane, and so on; the number of these shrinkages in course of one phase, as we remember, cannot be more than n, so that the phase is comprised of finitely many iterations; it follows that the number of phases indeed is infinite. On the other hand, at the end of a phase the method is at the critical point of the objective on the current working plane; since the method travels along the feasible set of (P), this point belongs to the corresponding face of G. There are finitely many working planes and, by assumption, finitely many belonging to G critical points of the objective on a fixed working plane, so that there are only finitely many "final positions" – the points x_t which could be final iterates of the method at a phase. Since the # of phases is infinite, it follows that the method is enforced to visit some of these final positions many times, which is impossible: from the description of the method it follows that the method is descent – it all the time reduces the objective – and, consequently, it never can visit the same point twice. ∎

9.1.2.4 Standard applications: Linear and Quadratic Programming

Problems most convenient for the Active Set scheme are those where the above "idealization" is in fact not an idealization – linearly constrained problems where the minimization of the objective over a given working plane is easy. There are two very important generic families of optimization problems fitting this requirement – Linear and Linearly Constrained Convex Quadratic Programming.

Linear Programming: not only the constraints in (P) are linear inequalities, but also the objective is linear. In this case the above Active Set scheme becomes as follows:

We start from a feasible solution x_0 and try to minimize our linear objective on the corresponding working plane S^0. If this plane is of positive dimension and the problem is below bounded (which we assume from now on), then the first step – minimization of the objective in a chosen descent direction of it in the working plane – will bring us to the boundary of the initial face G^0, the working plane will be shrunk, and so on, until the working plane will not become 0-dimensional, i.e., we will find ourselves at a vertex of the feasible domain[4)]. Starting with this moment, the procedure will be as follows. Our current working plane is a singleton which is a vertex of the feasible domain, and there are exactly n inequality constraints active at the iterate (recall that we are under the Regularity assumption!). Of course, this iterate is the critical point of the objective on the (0-dimensional) current working plane, so that we have relation (9.1.4). If current λ's are nonnegative, we are done – we have found a KKT point of the problem, i.e., a global solution to it (the problem is convex!). If there is a negative λ, it gives us extended by one dimension new working plane along with a new iterate in the corresponding one-dimensional face of the feasible domain. The subsequent step from this new iterate will bring us to a boundary point of this one-dimensional face, i.e., to a new vertex of the feasible domain, and will shrink the working plane by one dimension – it again will become 0-dimensional, and so on. As a result of two subsequent iterates

$$\text{vertex} \mapsto \text{better point on 1-dimensional face} \mapsto \text{better vertex}$$

we simply move from a vertex to a new vertex (linked with the previous one by an edge), each time improving the value of the objective, until we reach the optimal vertex. As you have already guessed, what we get is the usual Simplex method.

Linearly Constrained Convex Quadratic Programming. Now assume that the constraints in (P) are linear, and the objective

$$f(x) = \frac{1}{2}x^T Ax + b^T x$$

is a strongly convex quadratic form (A is a positive definite symmetric matrix). Problems of this type are extremely important by their own right; besides this, they are the auxiliary problems solved at each iteration of several general purpose optimization methods: the Sequential Quadratic Programming method for smooth nonconvex constrained optimization (this method is very popular now), Bundle methods for large-scale nonsmooth convex optimization, etc.

Quadratic Programming is a very convenient application field for the Active Set scheme, since here we can explicitly minimize the objective on the current working plane S_I. Indeed, representing S_I by the set of linear equations:

$$S_I = \{x \mid Px = p\},$$

P being a matrix of full row rank with the rows $(\nabla g_i)^T$, $i \in I$, we observe that the necessary (and also sufficient – f is convex!) condition for a point $x \in S$ to minimize f on S is the existence of multiplier vector λ of the same dimension as p such that

$$\begin{array}{rcl} Ax + b + P^T\lambda & = & 0 \\ Px & = & p \end{array} \tag{9.1.6}$$

(you can look at this system as at the KKT optimality condition or simply observe that $x \in S_I$ is the minimizer of f on S_I if and only if $\nabla f(x) = Ax + b$ is orthogonal to the linear subspace $\{h \mid Ph = 0\}$ of

[4)] we ignore the degenerate case when the objective is constant on a non-singleton face of G

the directions parallel to S_I, which in turn is the case if and only if $\nabla f(x)$ can be represented as a linear combination of rows of P, i.e., as $-P^T\lambda$ for some λ).

What we get is a square system of linear equations with unknowns x and λ, and you can easily prove that the matrix $\begin{pmatrix} A & P^T \\ P & 0 \end{pmatrix}$ of the system is nonsingular[5)]. Consequently, the system has a unique solution; x-component of this solution is exactly the minimizer of f on S.

With the possibility to find a minimizer of our strongly convex quadratic form on every working plane S_I in one step, just by solving the corresponding linear system, we can implement the Active Set scheme to find global minimizer of the function f on the polyhedral set G as follows:

> At an iteration t, we have a feasible solution x_{t-1} to the problem along with certain set J_{t-1} of indices of constraints active at x_{t-1} (this set can be the set I_{t-1} of indices of all constraints active at x_{t-1} or can be less than the latter set), and define the current working plane S^{t-1} as $S_{J^{t-1}}$.
>
> In course of iteration, we act as follows:
>
> 1) Find, as explained above, the minimizer x_t^+ of f on the working plane S^{t-1}, and check whether this point is feasible for the problem. If it is the case, we go to 2), otherwise to 3).
>
> 2) If x_t^+ is feasible for the problem, we check whether the Lagrange multipliers λ associated with this point as the solution to the problem
>
> $$\min_x \left\{ f(x) : x \in S^{t-1} \right\}$$
>
> are nonnegative; note that these Lagrange multipliers are given by the λ-part of the same system (9.1.6) which gives us x_t^+. If all the Lagrange multipliers are nonnegative, we terminate – x_t^+ is a KKT of (P), i.e., is a global solution to the problem (the problem is convex!). If some of λ's are negative, we choose one of these negative λ's, let the index of the corresponding constraint be i, and set
>
> $$x_t = x_t^+, \;\; J_t = J_{t-1} \backslash \{i\}.$$
>
> The iteration is complete.
>
> 3) If x_t^+ is infeasible for (P), we find the largest possible γ such that the point $x_t = x_{t-1} + \gamma(x_t^+ - x_t)$ is feasible for (P) and define J_t as the set of indices of all constraints which are active at x_t. The iteration is complete.

Using the reasoning outlined when motivating the Active Set scheme, one can easily verify that under our assumptions (feasibility of the problem, regularity of the system of (linear) constraints at every feasible point + strong convexity of the quadratic objective) the presented method terminates after finitely many steps with global solution to the problem. Let me stress that in spite of the fact that theoretically this Active Set algorithm for Linearly Constrained Convex Quadratic Programming, same as the Simplex method for Linear Programming, may be bad – perform an exponential in the number m of constraints amount of iterations – its practical behaviour typically is very nice, and there are strong experimental reasons to qualify this method as one of the best practical tools of Quadratic Programming.

We would also add that in principle the Active Set methods can be extended from the case of linear constraints to the general nonlinearly constrained case, but here they become too slow because of difficulties with travelling along "curved" working surfaces, which now replace "flat" working planes.

9.2 Penalty and Barrier Methods

Now let us look at the *penalty* and the *barrier* methods; as far as the underlying ideas are concerned, these methods implement the simplest approach to constrained optimization – approximate a constrained problem by unconstrained ones. Let us look how it is done.

[5)] here is the proof: assume that $Ah + P^T\nu = 0$, $Ph = 0$; we should prove that $h = 0, \nu = 0$. Multiplying the first equation by h^T and taking into account the second equation, we get $h^T Ah = 0$; since A is positive definite, we conclude that $h = 0$. Now the first equation reads $P^T\nu = 0$, and since P is of full row rank, $\nu = 0$.

9.2.1 The idea

The Penalty Scheme: equality constrained case. To get the idea of the construction, consider an equality constrained problem

$$\text{(ECP)} \quad \min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., m\} \quad [x \in \mathbf{R}^n].$$

In order to approximate this constrained problem by an unconstrained one, let us add to our objective a term which "penalizes" violation of constraints; the simplest term of this type is

$$\frac{1}{2}\rho \sum_{i=1}^{m} h_i^2(x),$$

where $\rho > 0$ is "penalty parameter". This term is zero on the feasible set and is positive outside this set; if ρ is large, then the penalizing term is large everywhere except tight neighbourhood of the feasible set.

Now let us add the penalty term to the objective. From the above discussion it follows that the resulting "combined objective"

$$f_\rho(x) = f(x) + \frac{1}{2}\rho \sum_{i=1}^{m} h_i^2(x) \tag{9.2.1}$$

possesses the following properties:

- at the feasible set, it coincides with the actual objective;
- for large ρ, is large outside tight neighbourhood of the feasible set (indeed, outside such a neighbourhood the penalty term becomes larger and larger as the penalty parameter grows, while the objective remains as it was).

From these properties it immediately follows that

$$\lim_{\rho \to \infty} f_\rho(x) = \begin{cases} f(x), & x \text{ feasible} \\ +\infty, & \text{otherwise} \end{cases};$$

Thus, we could say that the limit of f_ρ as $\rho \to \infty$ is the function taking values in the extended real axis (with $+\infty$ added) which coincides with f on the feasible set and is $+\infty$ otherwise this set; it is clear that *unconstrained* local/global minimizers of this limit are exactly the *constrained* local, respectively, global minimizers of f. This *exact coincidence* takes place only in the limit; we could, anyhow, expect that "close to the limit", for large enough values of the penalty parameter, the unconstrained minimizers of the penalized objective are close to the constrained minimizers of the actual objective. Thus, solving the unconstrained problem

$$\min_x f_\rho(x)$$

for large enough value of ρ, we may hope to get good approximations to the solutions of (ECP). As we shall see in the mean time, under mild regularity assumptions all these "could expect" and "may hope" indeed take place.

The Penalty Scheme: general constrained case. The idea of penalization can be easily carried out in the case when there are inequality constraints as well. Given a general type constrained problem

$$\text{(GCP)} \quad \min_x \{f(x) : h_i(x) = 0,\ i = 1, ..., m,\ g_j \le 0,\ j = 1, ..., k\},$$

we could penalize the inequality constraints by the term

$$\frac{1}{2}\rho \sum_{j=1}^{k} (g_j^+(x))^2,$$

where

$$a^+ = \max[a, 0] = \begin{cases} a, & a \geq 0 \\ 0, & a < 0 \end{cases}$$

is the "positive part" of a real a. The resulting penalty term is zero at any point where all the inequality constraints are satisfied and is positive (and proportional to ρ) at any point where at least one of the inequalities is violated.

Adding to the objective penalty terms for both equality and inequality constraints, we come to the penalized objective

$$f_\rho(x) = f(x) + \frac{1}{2}\rho \sum_{i=1}^{m} h_i^2(x) + \frac{1}{2}\rho \sum_{j=1}^{k} (g_j^+(x))^2; \tag{9.2.2}$$

same as above, we can expect that the unconstrained minimizers of the penalized objective approach the constrained minimizers of the actual objective as the penalty parameter goes to infinity. Thus, solutions of the unconstrained problem

$$\min_x f_\rho(x),$$

for large ρ, are good approximations to the solutions of (GCP).

The Barrier Scheme. The idea of the *barrier* methods is similar; the only difference is that instead of allowing violations of the constraints and penalizing these violations, we now prevent the constraints to be violated by a kind of interior penalty which blows up to infinity as a constraint is about to be violated. This "interior penalty" approach can be normally used in the case of *inequality constrained* problems with "full-dimensional" feasible set. Namely, consider an inequality constrained problem

$$\text{(ICP)} \quad \min_x \{f(x) : g_j(x) \leq 0,\ j = 1, ..., k\},$$

and assume that the feasible domain G of the problem is such that

- the interior int G of the domain G is nonempty, and every g_j is strictly negative in int G
- every point from G can be represented as the limit of a sequence of points from the interior of G

Assume also, just for the sake of simplicity, that G is bounded.

Given problem with the indicated properties, one can in many ways define an *interior penalty function* (also called a *barrier*) for the feasible domain G, i.e., a function F defined on the interior of G and such that

- F is continuously differentiable on int G
- $F(x_i) \to \infty$ for any sequence of points $x_i \in$ int G converging to a boundary point x of G

For example, one can set

$$F(x) = \sum_{j=1}^{k} \frac{1}{-g_j(x)}$$

(*the Carrol barrier*), or

$$F(x) = -\sum_{j=1}^{k} \ln(-g_j(x))$$

(*the logarithmic barrier*), or something else.

Now consider the following "aggregate":

$$F_\rho(x) = f(x) + \frac{1}{\rho} F(x),$$

where $\rho > 0$ is penalty parameter. The function F_ρ is well-defined and smooth on int G and goes to ∞ along every sequence of points from int G converging to a boundary point of G (indeed, along such a

sequence F possesses the required behaviour, while f remains bounded due to continuity of the objective). In particular, the level sets of F_ρ – the sets of the type

$$\{x \in \text{int } G \mid F_\rho(x) \le a\}$$

are closed[6)]. Since they are also bounded (recall that G is assumed to bounded), they are compact sets, and F_ρ, being continuous on such a compact set, attains its minimum on it; the corresponding minimizer clearly is a minimizer of F_ρ on int G as well.

Thus, for every positive ρ the function F_ρ attains its minimum on int G. At the same time, when the penalty parameter ρ is large, F_ρ "almost everywhere in int G" is "almost equal" to f – indeed, due to the factor $\frac{1}{\rho}$ at F in F_ρ, the contribution of the interior penalty term for large ρ is not negligible only in a thin, the smaller the larger is ρ, neighbourhood of the boundary. From this observation it is natural to guess (and it turns out to be indeed true) that the minimizers of F_ρ on int G are, for large ρ, close to the optimal set of (ICP) and could therefore be treated as good approximate solutions to (ICP).

Now, the problem

$$\min_x F_\rho(x)$$

is, formally, a constrained problem – since the domain of the objective is int G rather than entire $\mathbf{R}^n$. Nevertheless, we have basically the same possibilities to solve the problem as if it was unconstrained. Indeed, any *descent* (i.e., forming a sequence of iterates along which the objective never increases) method for unconstrained minimization, as applied to F_ρ and started at an interior point of G, never will come too close to the boundary of G (since, as we know, close to the boundary F_ρ is large, and along the trajectory of the method F_ρ is not greater than at the starting point). It means that the behaviour of the method as applied to F_ρ will, basically, be the same as if F_ρ was defined everywhere – the method simply will not feel that the objective is only partially defined. Thus, the barrier scheme in fact reduces the constrained minimization problem to an unconstrained one (or, better to say, allows to approximate the constrained problem by "essentially unconstrained" problem).

After the ideas of penalty and barrier schemes are outlined, let us come to more detailed investigation of the schemes.

9.2.2 Penalty methods

Let us investigate the penalty scheme in more details.

9.2.2.1 Convergence

The main questions we should focus on are

- Whether indeed unconstrained minimizers of the penalized objective f_ρ converge, as $\rho \to \infty$, to the solutions of the constrained problem?
- What are our possibilities to minimize the penalized objective?

For the sake of definiteness, let us focus on the case of equality constrained problem (ECP) (the results for the general case are similar).

The first – and very simple – statement is as follows:

[6)] indeed, to prove closedness of a level set, let it be called L, is the same as to prove that if $f_\rho(x_i) \le a < \infty$ for certain sequence of points $\{x_i\}$ converging to a point x, then $F_\rho(x) \le a$ (a closed set is by definition, the one which contains the limits of all converging sequences comprised of elements of the set). A priori x might be either an interior, or a boundary point of G. The second possibility should be excluded, since if it is the case, then, due to already indicated properties of F_ρ, it would be $F_\rho(x_i) \to \infty$ as $i \to \infty$, which is impossible, since F_ρ is above bounded on every level set. Thus, $x \in$ int G; but then F_ρ is continuous at x, and since $F_\rho(x_i) \le a$ and $x_i \to x$, $i \to \infty$, we get $F_\rho(x) \le a$, as required.

Theorem 9.2.1 *Let the objective f in problem (ECP) possess bounded level sets:*

$$f(x) \to \infty, \ |x| \to \infty,$$

and let (ECP) be feasible. Then, for any positive ρ, the set of global minimizers $X^*(\rho)$ of the penalized objective f_ρ is nonempty. Moreover, if X^* is the set of globally optimal solutions to (ECP), then, for large ρ, the set X^*_ρ is "close" to X^*: for any $\epsilon > 0$ there exists $\rho = \rho(\epsilon)$ such that the set $X^*(\rho)$, for all $\rho \geq \rho(\epsilon)$, is contained in ϵ-neighbourhood

$$X^*_\epsilon = \{x \mid \exists x^* \in X^* : \ |x - x^*| < \epsilon\}$$

of the optimal set of (ECP).

Proof. First of all, (ECP) is solvable. Indeed, let x_0 be a feasible solution to the problem, and let U be the corresponding level set of the objective:

$$U = \{x \mid \ f(x) \leq f(x_0)\}.$$

By assumption, this set is bounded and, due to continuity of f, is closed; therefore it is compact. Further, the feasible set S of the problem also is closed (since the constraints are continuous); consequently, the set

$$U_f = U \cap S$$

– the set of all feasible solutions not worse, in terms of the values of f, then the feasible solution x_0 – is bounded and closed, therefore is compact (and nonempty – it contains x_0). It is clear that the original problem is equivalent to the one of minimizing the objective over U_f, and the latter problem, being a problem of minimizing a continuous function on compact set, is solvable.

By similar reasons, every unconstrained problem

$$(\mathrm{P}_\rho) \quad \min_x f_\rho(x)$$

also is solvable. We claim that

- the optimal value f^*_ρ of (P_ρ) is not greater than the optimal value f^* in (ECP);
- if x^*_ρ is an optimal solution to (P_ρ), then

$$f(x^*_\rho) \leq f^*; \tag{9.2.3}$$

- the optimal set $X^*(\rho)$ of (P_ρ) is contained in U.

Indeed, if x^* is an optimal solution to (ECP), then

$$f_\rho(x^*) = f(x^*) = f^*,$$

so that the optimal value in (P_ρ) can be only $\leq$ the one in (ECP), which justifies the first claim. Further, due to nonnegativity of the penalty term we have

$$f(x^*_\rho) \leq f_\rho(x^*_\rho) = \min_x f_\rho(x) \leq f_\rho(x^*) = f^*,$$

which justifies the second claim. And this second claim immediately implies that $x^*_\rho \in U$ by construction of U.

Our observations immediately result in the desired conclusions. Indeed, we already have proved that $X^*(\rho)$ is nonempty, and all we need is to verify that, for large ρ, the set $X^*(\rho)$ is contained in tight neighbourhood of X^*. Assume that it is not the case: there exist positive ϵ and a sequence $\rho_i \to \infty$ such that $X^*(\rho_i)$ is not contained in X^*_ϵ, so that one can choose

$x_i^* \in X^*(\rho_i)$ in such a way that x_i^* is outside X_ϵ^*. According to the third claim, the points x_i^* belong to U and form therefore a bounded sequence. Passing to a subsequence, we may assume that x_i^* converge to certain point x as $i \to \infty$. W claim that x is an optimal solution to (ECP) (this will give us the desired contradiction: since $x_i^* \to x \in X^*$, the points x_i^* for all large enough i must be in X_ϵ^* – look at the definition of the latter set – and we have chosen x_i not to be in X_ϵ^*). To prove that x is an optimal solution to (ECP), we should prove that $f(x) \le f^*$ and that x is feasible. The first inequality is readily given by (9.2.3) – it should be satisfied for $x_\rho^* = x_i^*$, and the latter points converge to x; recall that f is continuous. Feasibility of x is evident: otherwise $h(x) \neq 0$ and, since $x_i^* \to x$ and h is continuous, for all large enough i one has

$$|h(x_i^*)| \ge a = \frac{1}{2}|h(x)| > 0,$$

whence for these i

$$f_{\rho_i}^* = f_{\rho_i}(x_i^*) = \frac{1}{2}\rho_i |h(x_i^*)|^2 + f(x_i^*) \ge \frac{a}{2}\rho_i + f(x_i^*) \to \infty, \ i \to \infty$$

(note that $\rho_i \to \infty$, while $f(x_i^*) \to f(x)$), which contradicts our first claim. ∎

The formulated theorem is not that useful: we could conclude something reasonable from its statement, if we were able to approximate the *global* solutions to (P_ρ), which is the case only when f_ρ is convex (as it, e.g., happens when f is convex and the equality constraints are linear). What we indeed need is a *local* version of the theorem. This version is as follows:

Theorem 9.2.2 *Let x^* be a nondegenerate locally optimal solution to (ECP), i.e., a solution where the gradients of the constraints are linearly independent and the Second Order Sufficient Optimality Condition from Lecture 7 is satisfied. Then there exists a neighbourhood V of x^* (an open set containing x^*) and $\bar\rho > 0$ such that for every $\rho \ge \bar\rho$ the penalized objective f_ρ possesses in V exactly one critical point $x^*(\rho)$. This point is a nondegenerate local minimizer of f_ρ and a minimizer of f_ρ in V, and $x^*(\rho) \to x^*$ as $\rho \to \infty$.*

Proof. The simplest way to prove the theorem is to reduce the situation to the case when the constraints are linear; this is the same approach we used to prove the Optimality Conditions in Lecture 7. The detailed proof is as follows (cf. the proof of the Optimality Conditions):

Since x^* is nondegenerate, the gradients of the constraints at x^* are linearly independent. From the appropriate version of the Implicit Function Theorem (we again use this magic Calculus tool) it follows that you can choose locally new coordinates y (nonlinearly related to the original ones!) in which our m constraints will be simply the first coordinate functions. Namely, there exist

- a neighbourhood V' of the point x^*
- a neighbourhood W' of the origin in $\mathbf{R}^n$
- a one-to-one mapping $x = X(y)$ from W' onto V' with inverse $y = Y(x)$

such that

- $X(\cdot)$ and $Y(\cdot)$ are continuously differentiable as many times as the constraints h (i.e., at least twice; recall that we once for ever restricted our considerations to problems with twice continuously differentiable data)
- $x^* = X(0)$ ("x^* in the coordinates y becomes the origin")
- $h_i(X(y)) \equiv y_i, \ i = 1, ..., m$ ("in y-coordinates the constraints become the first m coordinate functions").

Now let us pass in (ECP) from coordinates x to coordinates y, which results in the problem

$$(\mathrm{ECP}') \quad \min_y \{\phi(y) \equiv f(X(y)) : y_i \equiv h_i(X(y)) = 0,\ i = 1, ..., m\};$$

this, of course, makes sense only in the neighbourhood W' of the origin in y-variables.

1^0. We claim that $y^* = 0$ is nondegenerate solution to the problem (ECP$'$). Indeed, we should prove that this is a regular point for the constraints in the problem (which is evident) and that the Second Order Sufficient Optimality condition takes place at the point, i.e, there exist λ^* such that the Lagrange function

$$L'(y, \lambda) = \phi(y) + \sum_{i=1}^{m} \lambda_i y_i$$

satisfies

$$\nabla_y L'(y^*, \lambda^*) = 0$$

and

$$d^T \nabla_y^2 L'(y^*, \lambda^*) d > 0$$

for every nonzero vector d which is orthogonal to the gradients of the constraints of (ECP$'$). And what we know is that there exists λ^* which ensures similar properties of the Lagrange function

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

of the original problem at $x = x^*$. What we shall prove is that the latter λ^* satisfies all our needs in (ECP$'$) as well. Indeed, we clearly have

$$L'(y, \lambda^*) = L(X(y), \lambda^*),$$

whence, denoting by $X'(y)$ the $n \times n$ matrix of derivative of the mapping $X(\cdot)$ at y,

$$\nabla_y L'(y^*, \lambda^*) = [X'(0)]^T \nabla_x L(x^*, \lambda^*) = [X'(0)]^T 0 = 0$$

– $y^* = 0$ satisfies the first order part of the Optimality condition.

Further, $d^T \nabla_y [h(X(y))] = d^T [X'(y)]^T \nabla_x h(X(y))$, so that d is orthogonal to the gradients of $y_i \equiv h_i(X(y))$, $i = 1, ..., m$, if and only if $\bar{d} = X'(0)d$ is orthogonal to $\nabla h_i(x^*)$, $i = 1, ..., m$. Note also that $X'(0)$ is nonsingular (differentiate the identity $Y(X(y)) \equiv y$ to get $[X'(0)]^{-1} = Y'(x^*)$), so that d is nonzero if and only if $X'(0)d$ is.

Now let d be a nonzero vector which is orthogonal to the gradients of the constraints of (ECP$'$). As it was just explained, $\bar{d} = X'(0)d$ is a nonzero vector orthogonal to the gradients of h_i at x^*, and we have

$$d^T \nabla_y^2 L'(y^*, \lambda^*) d = d^T [\nabla_y^2|_{y=y^*} L(X(y), \lambda^*)] d =$$

[differentiating twice the superposition in direction d]

$$d^T [X'(0)]^T \nabla_x^2 L(x, \lambda^*) X'(0) d + [\nabla_x L(x^*, \lambda^*)]^T \frac{d^2}{dt^2}|_{t=0} X(td) =$$

[the second term is zero due to the origin of λ^*]

$$= \bar{d}^T \nabla_x^2 L(x^*, \lambda^*) \bar{d} > 0$$

[again due to the origin of λ^*], and we see that the second order part of the required conditions also is satisfied. ∎

2^0. Now note that (ECP′) is a problem with linear constraints, so that the Hessian with respect to y of the Lagrangian of the problem, independently of the values of the Lagrange multipliers, is $\nabla_y^2\phi(y)$. In particular, the "second-order part" of the fact that $y^* = 0$ is nondegenerate solution to (ECP′) simply says that $H = \nabla_y^2\phi(y^*)$ is positive definite on the plane

$$L = \{d \mid d_i = 0,\ i = 1, ..., m\}$$

(this the tangent plane at y^* to the feasible surface of the problem). This is the key argument in the proof of the following crucial fact:

(*) *function*

$$\phi_\rho(y) = \phi(y) + \frac{1}{2}\rho\sum_{i=1}^m y_i^2$$

– the penalized objective of (ECP′) – is, for large enough ρ, strictly convex in a properly chosen convex neighbourhood W of $y^* = 0$, i.e., there exists small enough convex neighbourhood $W \subset W'$ of $y^* = 0$ (simply an open ball of certain small enough positive radius) and $\rho^* > 0$ such that

$$\nabla_y^2\phi_\rho(y)$$

is positive definite whenever $y \in W$ and $\rho \geq \rho^*$.

The proof of (*) goes through the following simple

Lemma 9.2.1 *Let A be a symmetric $n \times n$ matrix and L be a linear subspace in $\mathbf{R}^n$, and let P be the orthoprojector on L. Assume that matrix A is positive definite on L:*

$$d^T Ad \geq \alpha|d|^2 \quad \forall d \in L$$

with certain $\alpha > 0$. Then there exists ρ^* such that the matrix

$$A + \rho(I - P)$$

is positive definite whenever $\rho \geq \rho^*$ and is such that

$$d^T(A + \rho(I - P))d \geq \frac{\alpha}{2}|d'|^2 + \frac{\rho}{2}|d''|^2 \quad \forall d \in \mathbf{R}^n,$$

$d' = Pd$ and $d'' = (I - P)d$ being the projections of d onto L and onto the orthogonal complement of L, respectively.

Moreover, ρ^* can be chosen depending only on α and on an upper bound β for the norm

$$|A| = \max_{d:|d|\leq 1} |Ad|$$

of the matrix A.

Proof. Let $\beta \geq |A|$ and $\gamma > 0$. We have

$$d^T(A + \rho(I - P))d = d^T Ad + \rho|d''|^2 = (d' + d'')^T A(d' + d'') + \rho|d''|^2 =$$

$$= (d')^T Ad' + 2(d')^T Ad'' + (d'')^T Ad'' + \rho|d''|^2 \geq$$

[since $(d')^T Ad' \geq \alpha|d'|^2$ and, by Cauchy's inequality,

$$|2(d')^T Ad''| = 2|d'||Ad''| \leq 2|A||d'||d''| \leq 2\beta|d'||d''| \leq \frac{\beta}{\gamma}|d'|^2 + \beta\gamma|d''|^2$$

(note that $|2uv| \leq \gamma^{-1}u^2 + \gamma v^2$ – this is nothing but the inequality between the arithmetic and the geometric mean)]

$$\geq \alpha|d'|^2 - \frac{\beta}{\gamma}|d'|^2 - \beta\gamma|d''|^2 + \rho|d''|^2.$$

Setting here $\gamma = \frac{2\beta}{\alpha}$, we get

$$d^T(A + \rho(I-P))d \geq \frac{\alpha}{2}|d'|^2 + [\rho - \frac{2\beta^2}{\alpha}]|d''|^2;$$

choosing finally $\rho^* = \frac{d4\beta^2}{\alpha}$ and assuming $\rho \geq \rho^*$, so that $\rho - \frac{2\beta^2}{\alpha} \geq \frac{\rho}{2}$, we come to

$$d^T(A + \rho(I-P)d \geq \frac{\alpha}{2}|d'|^2 + \frac{\rho}{2}|d''|^2$$

for all $\rho \geq \rho^*$. ■

Now we are ready to prove (*). Indeed, since $\nabla^2\phi(y)$ is continuous in y and $\nabla^2\phi(y^*)$ is positive definite on

$$L = \{d \mid d_i = 0, i = 1, ..., m\},$$

we could choose

- small enough ball W centered at $y^* = 0$ and contained in W'
- small enough positive α
- large enough positive β

such that

$$d^T[\nabla^2\phi(y)]d \geq \alpha|d|^2,\ y \in W, d \in L,$$

P being the orthoprojector on L, and

$$|\nabla^2\phi(y)| \leq \beta,\ y \in W.$$

Now note that

$$\nabla_y^2\phi_\rho(y) \equiv \nabla_y^2\left[\phi(y) + \frac{\rho}{2}\sum_{i=1}^m y_i^2\right] = \nabla^2\phi(y) + \rho(I-P),$$

P being the orthoprojector onto L. According to Lemma 9.2.1, there exists $\rho^* > 0$ such that all the matrices

$$\nabla^2\phi_\rho(y)$$

corresponding to $y \in W$ and $\rho \geq \rho^*$ are positive definite, moreover, satisfy

$$d^T[\nabla^2\phi_\rho(y)]d \geq \frac{\alpha}{2}|d'|^2 + \frac{\rho}{2}|d''|^2,\ \ d' = Pd,\ d'' = (I-P)d. \tag{9.2.4}$$

Thus, whenever $\rho \geq \rho^*$, the Hessian of the function ϕ_ρ is positive definite in W, so that ϕ_ρ is convex in W.

4^0. Let $\rho \geq \rho^*$. From (9.2.4) it immediately follows that

$$\phi_\rho(y) \geq \phi_\rho(0) + y^T\nabla_y\phi(0) + \frac{\alpha}{4}|y'|^2 + \frac{\rho}{4}|y''|^2,\ \ y' = Py,\ y'' = (I-P)y''. \tag{9.2.5}$$

The gradient of ϕ_ρ at $y^* = 0$ is, first, independent of ρ (it is simply the gradient of ϕ at the point) and, second, is orthogonal to L, as it is given by the "first order part" of the fact that $y^* = 0$ is a nondegenerate solution to (ECP′). Consequently, (9.2.5) can be rewritten as

$$\phi_\rho(y) \geq \phi_\rho(0) + (y'')^T g + \frac{\alpha}{4}|y'|^2 + \frac{\rho}{4}|y''|^2, \tag{9.2.6}$$

$g = \nabla_y \phi(y^*)$ being a once for ever fixed vector. From this relation it easily follows[7)]

(**) *there exists $\bar\rho \geq \rho^*$ such that $\phi_\rho(y) > \phi_\rho(0)$ whenever $\rho \geq \bar\rho$ and y is a boundary point of the ball W.*

Now let $\rho \geq \bar\rho$. The function ϕ_ρ, being continuous on the closure of W (which is a closed ball, i.e., a compact set), attains its minimum on $\mathrm{cl}\, W$, and, due to strong convexity of the function, the minimizer, let it be called $y^*(\rho)$, is unique. By (**), this minimizer cannot be a boundary point of $\mathrm{cl}\, W$ – on the boundary ϕ_ρ is greater than at $y^* = 0$, i.e., it is a point from W. Since ϕ_ρ is smooth and $y^*(\rho) \in W$, $y^*(\rho)$ is a critical point of ϕ_ρ. There is no other critical point of ϕ_ρ in W, since ϕ_ρ is convex and therefore a critical point of the function is its minimizer on $\mathrm{cl}\, W$, and we know that such a minimizer is unique. Note that $y^*(\rho)$ is a nondegenerate local minimizer of ϕ_ρ, since ϕ_ρ is with positive definite Hessian. Last, $y^*(\rho) \to y^* = 0$ as $\rho \to \infty$. Indeed, we have

$$\phi_\rho(y^*(\rho)) \leq \phi_\rho(0),$$

whence, denoting $y_L(\rho) = Py^*(\rho)$, $y_{L^\perp}(\rho) = (I - P)y^*(\rho)$ and applying (9.2.6),

$$g^T y_{L^\perp}(\rho) + \frac{\alpha}{4}|y_L(\rho)|^2 + \frac{\rho}{4}|y_{L^\perp}(\rho)|^2 \leq 0,$$

or, with the Cauchy inequality,

$$|g||y_{L^\perp}(\rho)| \geq \frac{\alpha}{4}|y_L(\rho)|^2 + \frac{\rho}{4}|y_{L^\perp}(\rho)|^2.$$

From this inequality it immediately follows that $|y_{L^\perp}(\rho)| \to 0$ as $\rho \to \infty$ (why?), and with this observation the same inequality results also in $|y_L(\rho)| \to 0$, $\rho \to \infty$, so that indeed $y^*(\rho) \to y^* = 0$, $\rho \to \infty$.

5^0. Thus, we have established the statement we are going to prove – but for the "locally equivalent to (ECP)" problem (ECP′) rather than for the actual problem of interest: we have pointed out a neighbourhood W of the point y^* such that the penalized objective ϕ_ρ of (ECP′), for all large enough ρ, has in this neighbourhood exactly one critical point, which is a nondegenerate minimizer of ϕ_ρ in the neighbourhood; and as $\rho \to \infty$, this critical point $y^*(\rho)$ converges to $y^* = 0$. Now let us take the image V of W under our substitution of variables mapping $y \mapsto X(y)$. We will get a neighbourhood of x^*, and in this neighbourhood we clearly have

$$f_\rho(x) = \phi_\rho(Y(x)).$$

Now, Y is one-to-one differentiable mapping of V onto W with differentiable inverse X; it immediately follows that a point x is a critical point (or a minimizer) of f_ρ in V if and only if $Y(x)$ is a critical point, respectively, a minimizer of ϕ_ρ in W; in particular, for $\rho \geq \bar\rho$ f_ρ

[7)] here is the derivation: we should prove that

$$(y'')^T g + \frac{\alpha}{4}|y'|^2 + \frac{\rho}{4}|y''|^2 > 0$$

whenever y is on the boundary of W and ρ is large enough; recall that W is centered at the origin ball of certain radius $r > 0$. Denoting $s = |y''|^2$ and taking into account that $|y'|^2 = r^2 - s$ and $(y'')^T g \geq -cs^{1/2}$ by Cauchy's inequality, we reduce the problem in question to the following one: prove that

$$(!) \qquad \min_{0 \leq s \leq r^2} \theta_\rho(s),\ \theta_\rho(s) = \left[\frac{\alpha}{4}r^2 + \frac{\rho - \alpha}{4}s - c\sqrt{s}\right]$$

is positive, provided that ρ is large enough. This is evident (split the entire segment $[0, r^2]$ where s varies into the segment Δ where $cs^{1/2} \leq \frac{\alpha}{8}r^2$ – in this segment $\theta_\rho(s)$ is positive whenever $\rho > \alpha$ – and the complementary segment Δ', and note that in this complementary segment s is bounded away from zero and therefore θ_ρ for sure is positive for all large enough values of ρ.

indeed possesses unique critical point $x^*(\rho) = X(y^*(\rho))$ in V, and this is the minimizer of f_ρ in V. As $\rho \to \infty$, we have $y^*(\rho) \to y^* = 0$, whence $x^*(\rho) \to X(0) = x^*$. The only property of $x^*(\rho)$ we did not verify so far is that it is *nondegenerate* local minimizer of f_ρ, i.e., that $\nabla_x^2 f_\rho(x^*(\rho))$ is positive definite; we know that it is the case for ϕ_ρ and $y^*(\rho)$, but our substitution of variables is *nonlinear* and therefore, generally speaking, does not preserve positive definiteness of Hessians. Fortunately, it does preserve positive definiteness of the Hessians *taken at critical points*: if $Y(x)$ is twice continuously differentiable mapping with differentiable inverse and ψ is such that $\nabla_y \psi(\bar{y}) = 0$, $\bar{y} = Y(\bar{x})$, then, as it is immediately seen, the Hessian of the composite function $g(x) = \psi(Y(x))$ at the point $\bar{x}$ is given by

$$\nabla_x^2 g(\bar{x}) = [Y'(\bar{x})]^T \nabla_y^2 \psi(\bar{y}) Y'(\bar{x})$$

(in the general case, there are also terms coming from the first order derivative of ψ at $\bar{y}$ and second order derivatives of $Y(\cdot)$, but in our case, when $\bar{y}$ is a critical point of ψ, these terms are zero), so that $\nabla_x^2 g(\bar{x})$ is positive definite if and only if $\nabla_y^2 \psi(\bar{y})$ is so. Applying this observation to $\psi = \phi_\rho$ and $\bar{y} = y^*(\rho)$, we obtain the last fact we need – nondegeneracy of $x^*(\rho)$ as an unconstrained local minimizer of f_ρ. ■

9.2.2.2 Properties of the path $x^*(\rho)$

In this subsection we assume that we are in situation of Theorem 9.2.2; our goal is to establish several useful properties of the given by the Theorem *path of unconstrained minimizers* $x^*(\rho), \rho \geq \bar{\rho}$ of the penalized objective in the neighbourhood V of x^*. The properties are as follows

- Let
$$X^+(\rho) = \{x \in V \mid |h(x)| \leq |h(x^*(\rho))|\}.$$
Then
$$x^*(\rho) \in \operatorname*{Argmin}_{x \in X^+(\rho)} f(x) \tag{9.2.7}$$
("$x^*(\rho)$ minimizes f on the set of all those points from V where the constraints are violated at most as they are violated at $x^*(\rho)$").

Indeed, $x^*(\rho)$ evidently belongs to $X^+(\rho)$; if there were a point x in $X^+(\rho)$ with $f(x) < f(x^*(\rho))$, we would have

$$f_\rho(x) = f(x) + \frac{\rho}{2}|h(x)|^2 <$$

[since $f(x) < f(x^*(\rho))$ and $|h(x)| \leq |h(x^*(\rho))|$ due to $x \in X^+(\rho)$]

$$< f(x^*(\rho)) + \frac{\rho}{2}|h(x^*(\rho))|^2 = f_\rho(x^*(\rho)).$$

The resulting inequality is impossible, since $x^*(\rho)$ minimizes f_ρ in V.

- [Monotonicity of the optimal value of the penalized objective] The optimal (in V) values $f_\rho(x^*(\rho))$ of the penalized objectives do not decrease with ρ.

  Indeed, if $\rho \leq \rho'$, then clearly $f_\rho(x) \leq f_{\rho'}(x)$ everywhere in V, and consequently the same inequality holds for the minimal values of the functions.

- [Monotonicity of violations of the constraints] The quantities
$$v(\rho) = |h(x^*(\rho))|$$
do not increase with ρ ("the larger is the penalty parameter, the less are violations of the constraints at the solution to the penalized problem")

Indeed, assume that $\rho' > \rho''$, and let $x' = x^*(\rho')$, $x'' = x^*(\rho'')$. We have

$$[f_{\rho'}(x'') \equiv] \quad f(x'') + \frac{\rho'}{2}|h(x'')|^2 \geq f(x') + \frac{\rho'}{2}|h(x')|^2 \quad [\equiv f_{\rho'}(x')]$$

and, similarly,

$$f(x') + \frac{\rho''}{2}|h(x')|^2 \geq f(x'') + \frac{\rho''}{2}|h(x'')|^2.$$

Taking sum of these inequalities, we get

$$\frac{\rho' - \rho''}{2}|h(x'')|^2 \geq \frac{\rho' - \rho''}{2}|h(x')|^2$$

whence, due to $\rho' > \rho''$, $v(\rho'') = |h(x'')| \geq |h(x')| = v(\rho')$, as required.

- [Monotonicity of the actual objective] The values of the actual objective f along the path $x^*(\rho)$ do not decrease with ρ.

  Indeed, we already know that

  $$f(x^*(\rho)) = \min_{x: x \in V, |h(x)| \leq v(\rho)} f(x).$$

  According to the previous statement, the sets in the right hand side over which the minimum is taken do not increase with ρ, and, consequently, the minimal value of f over these sets does not decrease with ρ.

The indicated properties demonstrate the following nice behaviour of the path $x^*(\rho)$: as the penalty parameter grows, the path approaches the constrained minimizer x^* of (ECP); the values of the objective along the path are always better (more exactly, not worse) than at x^* and increase (actually, do not decrease) with ρ, approaching the optimal value of the constrained problem from the left. Similarly, the violation of constraints becomes smaller and smaller as ρ grows and approaches zero. In other words, for every finite value of the penalty parameter it turns out to be profitable to violate constraints, getting, as a result, certain progress in the actual objective; and as the penalty parameter grows, this violation, same as progress in the objective, monotonically goes to zero.

An additional important property of the path is as follows:

- [Lagrange multipliers and the path] The quantities

  $$\lambda_i(\rho) = \rho h_i(x^*(\rho))$$

  tend, as $\rho \to \infty$, to the optimal Lagrange multipliers λ^* associated with x^*.

  Indeed, $x^*(\rho)$ is a critical point of f_ρ, whence

  $$\nabla_x f_\rho(x^*(\rho)) = \nabla_x f(x^*(\rho)) + \sum_{i=1}^m \lambda_i(\rho) \nabla_x h(x^*(\rho)) = 0.$$

  Thus, $\lambda_i(\rho)$ are the coefficients in the representation of the vector $\psi(\rho) \equiv -\nabla_x f(x^*(\rho))$ as a linear combination of the vectors $\psi_i(\rho) = \nabla_x h_i(x^*(\rho))$. As we know, $x^*(\rho) \to x^*$ as $\rho \to \infty$, so that $\psi(\rho) \to \psi \equiv \nabla_x f(x^*)$ and $\psi_i(\rho) \to \psi_i \equiv \nabla_x h_i(x^*)$. Since the vectors $\psi_1, ..., \psi_m$ are linearly independent, from the indicated convergencies it follows (why?) that $\lambda_i(\rho) \to \lambda_i^*$, $\rho \to \infty$, where λ_i^* are the (uniquely defined) coefficients in the representation of $-\psi$ as a linear combination of ψ_i, i.e., are the Lagrange multipliers.

Remark 9.2.1 Similar results and properties take place also for the penalty method as applied to the general type constrained problem (GCP).

Remark 9.2.2 The quadratic penalty term we used is not, of course, the only option; for (ECP), we could use penalty term of the form

$$\rho\Phi(h(x))$$

as well, with smooth function Φ which is zero at the origin and positive outside the origin (in our considerations, Φ was set to $\frac{1}{2}|u|^2$); similar generalizations are possible for (GCP). The results for these more general penalties, under reasonable assumptions on Φ, would be similar to those as for the particular case we have considered.

Advantages and drawbacks. Now it is time to look what are our abilities to solve the unconstrained problems

$$(\mathrm{P}_\rho) \quad \min_x f_\rho(x)$$

which, as we already know, for large ρ are good approximations of the constrained problem in question. *In principle* we can solve these problems by any one of unconstrained minimization methods we know, and this is definitely a great advantage of the approach.

There is, anyhow, a severe weak point of the construction – to approximate well the constrained problem by unconstrained one, we *must* deal with large values of the penalty parameter, and this, as we shall see in a while, unavoidably makes the unconstrained problem (P_ρ) *ill-conditioned* and thus – very difficult for any unconstrained minimization methods sensitive to the conditioning of the problem. And all the methods for unconstrained minimization we know, except, possibly, the Newton method, are "sensitive" to conditioning (e.g., in the Gradient Descent the number of steps required to achieve an ϵ-solution is, asymptotically, inverse proportional to the condition number of the Hessian of objective at the optimal point). Even the Newton method, which does not react on the conditioning explicitly – it is "self-scaled" – suffers a lot as applied to an ill-conditioned problem, since here we are enforced to invert ill-conditioned Hessian matrices, and this, in actual computations with their rounding errors, causes a lot of troubles. The indicated drawback – ill-conditioness of auxiliary unconstrained problems – is the main disadvantage of the "straightforward" penalty scheme, and because of it the scheme is not that widely used now and is in many cases replaced with more smart modified Lagrangian scheme.

It is time now to justify the above claim that problem (P_ρ) is, for large ρ, ill-conditioned. Indeed, assume that we are in the situation of Theorem 9.2.2, and let us compute the Hessian H_ρ of f_ρ at the point $x = x^*(\rho)$. The computation yields

$$H_\rho = \left[\nabla_x^2 f(x) + \sum_{i=1}^m [\rho h_i(x)]\nabla_x^2 h_i(x)\right] + \rho[\nabla_x h(x)]^T[\nabla_x h(x)].$$

We see that H_ρ is comprised of two terms: the matrix in the brackets, let it be called L_ρ, and the proportional to ρ matrix $\rho M_\rho \equiv [\nabla_x h(x)]^T[\nabla_x h(x)]$. When $\rho \to \infty$, then, as we know, $x = x^*(\rho)$ converges to x^* and $\rho h_i(x^*)$ converge to the Lagrange multipliers λ_i^* of (ECP), so that L_ρ possesses quite respectable limit L, namely, the Hessian of the Lagrange function $\nabla_x^2 L(x^*, \lambda^*)$. The matrix M_ρ also possesses limit, namely,

$$M = [\nabla_x h(x^*)]^T[\nabla_x h(x^*)];$$

this limit, as it is clearly seen, is a matrix which vanishes on the tangent at x^* plane T to the feasible surface of the problem and is nondegenerate on the orthogonal complement $T^\perp$ to this tangent plane. Since M_ρ is symmetric, both T and $T^\perp$ are invariant for M, and M possesses $n - m$ eigenvectors with zero eigenvalues – these vectors span T – and m eigenvectors with positive eigenvalues – these latter vectors span $T^\perp$. Since

$$H_\rho = L_\rho + \rho M_\rho,$$

we conclude that the spectrum of H_ρ, for large ρ, is as follows: there are $n - m$ eigenvectors "almost in T" with the eigenvalues "almost equal" to those of the reduction of L onto T;

since L is positive definite on T, these eigenvalues are positive reals. Now, H_ρ possesses m eigenvectors "almost orthogonal" to T with eigenvalues "almost equal" to ρ times the nonzero eigenvalues of M. Thus, excluding trivial cases $m = 0$ (no constraints at all) and $m = n$ (locally unique feasible solution x^*), the eigenvalues of H_ρ form two groups – group of $n-m$ asymptotically constant positive reals and group of m reals asymptotically proportional to ρ. We conclude that the condition number of H_ρ is of order of ρ, as it was claimed.

9.2.3 Barrier methods

"Classical" barrier scheme. The situation with the *barrier* (interior penalty) methods in their "classical" from outlined in Section 9.2.1 is very similar to the one with penalty methods. It indeed is true (and is easily verified) that the solutions to the modified problem

$$(\mathrm{P}_\rho) \quad \min_x F_\rho(x) \equiv f(x) + \frac{1}{\rho} F(x),$$

F being the interior penalty for the feasible domain of (ICP), converge to the optimal set of the problem:

Theorem 9.2.3 *Let F be an interior penalty function for the feasible domain G of (ICP), and assume that the feasible domain is bounded and is the closure of its interior* int G. *Then the set $X^*(\rho)$ of minimizers of F_ρ on* int G *is nonempty, and these sets converge, as $\rho \to \infty$, to the optimal set X^* of (ICP): for every $\epsilon > 0$ there exists $\bar\rho$ such that $X^*(\rho)$, for all $\rho \geq \bar\rho$, is contained in the ϵ-neighbourhood*

$$X^*_\epsilon = \{x \in G \mid \exists x^* \in X^* : |x - x^*| < \epsilon\}$$

of the optimal set of (ICP).

Proof is completely similar to that one of Theorem 9.2.1. First of all, X^* is nonempty (since f is continuous and the feasible domain is bounded and closed and is therefore a compact set). The fact that $X^*(\rho)$ are nonempty for all positive ρ was proved in Section 9.2.1. To prove that $X^*(\rho)$ is contained, for large ρ, in a tight neighbourhood of X^*, let us act as follows. Same as in the proof of Theorem 9.2.1, it suffices to lead to a contradiction the assumption that there exists a sequence $\{x_i \in X^*(\rho_i)\}$ with $\rho_i \to \infty$ which converges to a point $x \in G \backslash X^*$. Assume that it is the case; then $f(x) > \min\limits_G f + \delta$ with certain positive δ. Let x^* be a global minimizer of f on G. Since G is the closure of int G, x^* can be approximated, within an arbitrarily high accuracy, by points from int G, and since f is continuous, we can find a point $x' \in$ int G such that

$$f(x') \leq f(x^*) + \frac{\delta}{2}.$$

We have

$$F_{\rho_i}(x_i) = \min_{x \in \mathrm{int}\, G} F_{\rho_i}(x) \leq F_{\rho_i}(x'), \tag{9.2.8}$$

whence

$$f(x_i) + \frac{1}{\rho_i} F(x_i) \leq f(x') + \frac{1}{\rho_i} F(x').$$

Since F is a barrier for *bounded* domain G, F is below bounded on int G (since it attains its minimum on int G – use the reasoning from Section 9.2.1 for the case of $f \equiv 0$). Thus, $F(x) \geq a > -\infty$ for all $x \in$ int G, and therefore (9.2.8) implies that

$$f(x_i) \leq f(x') + \frac{1}{\rho_i} F(x') - \frac{1}{\rho_i} a.$$

As $i \to \infty$, the right hand side in this inequality tends to $f(x') \leq f(x^*) + \frac{\delta}{2}$, and the left hand side tends to $f(x) \geq f(x^*) + \delta$; since $\delta > 0$, we get the desired contradiction. ∎

If I was writing this lecture in 1980's, we would proceed with the statements similar to the one of Theorem 9.2.2 and those on behaviour of the path of minimizers of F_ρ and conclude with the same laments "all this is fine, but the problems of minimization of F_ρ normally (when the solution to the original problem is on the boundary of G; otherwise the problem actually is unconstrained) become the more ill-conditioned the larger is ρ, so that the difficulties of their numerical solution grow with the penalty parameter". When indeed writing this lecture, I am saying something quite opposite: *there exists important situations when the difficulties in numerical minimization of F_ρ do not increase with the penalty parameter, and the overall scheme turns out to be theoretically efficient and, moreover, the best known so far.* This change in evaluation of the scheme is the result of recent "interior point revolution" in Optimization which we have already mentioned in Lecture 10. Those interested may get some impression of essence of the matter from the below non-obligatory part of the lecture; please take into account that the prerequisite for reading it is non-obligatory Section 6.2.4 from Lecture 9.

9.2.3.1 Self-concordant barriers and path-following scheme

Assume from now on that our problem (ICP) is convex (the revolution we are speaking about deals, at least directly, with Convex Programming only). It is well-known that convex program (ICP) can be easily rewritten as a program with *linear objective*; indeed, it suffices to extend the vector of design variables by one variable, let it be called t, more and to rewrite (ICP) as the problem

$$\min_{t,x} \{t : g_j(x) \le 0,\ j = 1, ..., k, f(x) - t \le 0\}.$$

The resulting problem still is convex and has linear objective.

To save notation, assume that the outlined conversion is carried out in advance, so that already the original problem has linear objective and is therefore of the form

$$\text{(P)} \quad \min_x \left\{ f(x) \equiv c^T x : x \in G \subset \mathbf{R}^n \right\}.$$

Here the feasible set G of the problem is convex (we are speaking about convex programs!); we also assume that it is *closed, bounded and possesses a nonempty interior.*

Our abilities to solve (P) efficiently by an interior point method depend on our abilities to point out a "good" interior penalty F for the feasible domain. What we are interested in is a ϑ-self-concordant barrier F; the meaning of these words is given by the following

Definition 9.2.1 [Self-concordant barrier] *Let $\vartheta \ge 1$. We say that a function F is ϑ-self-concordant barrier for the feasible domain D of problem (P), if*

- *F is self-concordant function on* int G *(Section 6.2.4.2, Lecture 4), i.e., three times continuously differentiable convex function on* int G *possessing the barrier property (i.e., $F(x_i) \to \infty$ along every sequence of points $x_i \in$ int G converging to a boundary point of G) and satisfying the differential inequality*

$$|\frac{d^3}{dt^3}|_{t=0} F(x + th)| \le 2 \left[h^T F''(x) h\right]^{3/2} \quad \forall x \in \text{int } G \ \forall h \in \mathbf{R}^n;$$

- *F satisfies the differential inequality*

$$|h^T F'(x)| \le \sqrt{\vartheta} \sqrt{h^T F''(x) h} \quad \forall x \in \text{int } G \ \forall h \in \mathbf{R}^n. \tag{9.2.9}$$

An immediate example is as follows (cf. "Raw materials" in Section 6.2.4.4, Lecture 4):

Example 9.2.1 [Logarithmic barrier for a polytope] *Let*

$$G = \{x \in \mathbf{R}^n \mid a_j^T x \leq b_j,\ j = 1, ..., m\}$$

be a polytope given by a list of linear inequalities satisfying the Slater condition (i.e., there exists $\bar{x}$ such that $a_j^T \bar{x} < b_j$, $j = 1, ..., m$). Then the function

$$F(x) = -\sum_{j=1}^{m} \ln(b_j - a_j^T x)$$

is an m-self-concordant barrier for G.

In the mean time, we shall justify this example (same as shall consider the crucial issue of *how* to find a self-concordant barrier for a given feasible domain). For the time being, let us focus on another issue: how to solve (P), *given a ϑ-self-concordant barrier for the feasible domain of the problem.*

What we intend to do is to use the *path-following scheme* associated with the barrier – certain very natural implementation of the barrier method.

9.2.3.2 Path-following scheme

When speaking about the barrier scheme, we simply claimed that the minimizers of the aggregate F_ρ approach, as $\rho \to \infty$, the optimal set of (P). The immediate recommendation coming from this claim could be: choose "large enough" value of the penalty and solve, for the chosen ρ, the problem (P_ρ) of minimizing F_ρ, thus coming to a good approximate solution to (P). It makes, anyhow, sense to come to the solution of (P_ρ) *gradually*, by solving sequentially problems (P_{ρ_i}) along an increasing sequence of values of the penalty parameter. Namely, assume that the barrier F we use is nondegenerate, i.e., $F''(x)$ is positive definite at every point $x \in \text{int } G$ (note that this indeed is the case when F is self-concordant barrier for a bounded feasible domain, see Proposition 6.2.2.(i)). Then the optimal set of F_ρ for every positive ρ is a singleton (we already know that it is nonempty, and the uniqueness of the minimizer follows from convexity and nondegeneracy of F_ρ). Thus, we have a *path*

$$x^*(\rho) = \operatorname*{argmin}_{\text{int } G} F_\rho(\cdot);$$

as we know from Theorem 9.2.3, this path converges to the optimal set of (P) as $\rho \to \infty$; besides this, it can be easily seen that the path is continuous (even continuously differentiable) in ρ. In order to approximate $x^*(\rho)$ with large values of ρ via the path-following scheme, we *trace the path* $x^*(\rho)$, namely, generate sequentially approximations $x(\rho_i)$ to the points $x^*(\rho_t)$ along certain diverging to infinity sequence $\rho_0 < \rho_1 < ...$ of values of the parameter. This is done as follows:

given "tight" approximation $x(\rho_t)$ to $x^*(\rho_t)$, we update it into "tight" approximation $x(\rho_{t+1})$ to $x^*(\rho_{t+1})$ as follows:

- *first, choose somehow a new value $\rho_{t+1} > \rho_t$ of the penalty parameter*
- *second, apply to the function $F_{\rho_{t+1}}(\cdot)$ a method for unconstrained minimization started at $x(\rho_t)$, and run the method until closeness to the new target point $x^*(\rho_{t+1})$ is restored, thus coming to the new iterate $x(\rho_{t+1})$ close to the new target point of the path.*

All this is very close to what we did when tracing feasible surface with the Gradient Projection scheme; our hope is that since $x^*(\rho)$ is continuous in ρ and $x(\rho_t)$ is "close" to $x^*(\rho_t)$, for "not too large" $\rho_{t+1} - \rho_t$ the point $x(\rho_t)$ will be "not too far" from the new target point $x^*(\rho_{t+1})$, so that the unconstrained minimization method we use will quickly restore closeness to the

new target point. With this "gradual" movement, we may hope to arrive near $x^*(\rho)$ with large ρ faster than by attacking the problem (P_ρ) directly.

All this was known for many years; and the progress during last decade was in transforming these qualitative ideas into exact quantitative recommendations.

Namely, it turned out that

- **A.** The best possibilities to carry this scheme out are when the barrier F is ϑ-self-concordant; the less is the value of ϑ, the better;
- **B.** The natural measure of "closeness" of a point $x \in \mathrm{int}\, G$ to the point $x^*(\rho)$ of the path is the Newton decrement of the self-concordant function

$$\Phi_\rho(x) = \rho F_\rho(x) \equiv \rho c^T x + F(x)$$

at the point x, i.e., the quantity

$$\lambda(\Phi_\rho, x) = \sqrt{[\nabla_x \Phi_\rho(x)]^T [\nabla_x^2 \Phi_\rho(x)]^{-1} \nabla_x \Phi_\rho(x)}$$

(cf. Proposition 6.2.2.(iii)). More specifically, the notion "x is close to $x^*(\rho)$" is convenient to define as the relation

$$\lambda(\Phi_\rho, x) \le 0.05 \tag{9.2.10}$$

(in fact, 0.05 in the right hand side could be replaced with arbitrary absolute constant < 1, with slight modification of subsequent statements; we choose this particular value for the sake of simplicity)

Now, what do all these words "the best possibility" and "natural measure" actually mean? It is said by the following two statements.

- **C.** *Assume that x is close, in the sense of (9.2.10), to a point $x^*(\rho)$ of the path $x^*(\cdot)$ associated with a ϑ-self-concordant barrier for the feasible domain G of problem (P). Let us increase the parameter ρ to the larger value*

$$\rho^+ = \left(1 + \frac{0.08}{\sqrt{\vartheta}}\right) \rho \tag{9.2.11}$$

and replace x by its damped Newton iterate (cf. (6.2.13), Lecture 4)

$$x^+ = x - \frac{1}{1 + \lambda(\Phi_{\rho^+}, x)} [\nabla_x^2 \Phi_{\rho^+}(x)]^{-1} \nabla_x \Phi_{\rho^+}(x). \tag{9.2.12}$$

Then x^+ is close, in the sense of (9.2.10), to the new target point $x^*(\rho^+)$ of the path.

C. says that we are able to trace the path (all the time staying close to it in the sense of **B.**) increasing the penalty parameter linearly in the ratio $(1 + 0.08\vartheta^{-1/2})$ and accompanying each step in the penalty parameter by a single Newton step in x. And why we should be happy with this, it is said by

- **D.** *If x is close, in the sense of (9.2.10), to a point $x^*(\rho)$ of the path, then the inaccuracy, in terms of the objective, of the point x as of an approximate solution to (P) is bounded from above by $2\vartheta\rho^{-1}$:*

$$f(x) - \min_{x \in G} f(x) \le \frac{2\vartheta}{\rho}. \tag{9.2.13}$$

D. says that the inaccuracy of the iterates $x(\rho_i)$ formed in the above path-following procedure goes to 0 as $1/\rho_i$, while **C.** says that we are able increase ρ_i linearly, at the cost of a single Newton step per each updating of ρ. Thus, we come to the following

Theorem 9.2.4 *Assume that we are given*

- (i) *ϑ-self-concordant barrier F for the feasible domain G of problem (P)*
- (ii) *starting pair (x_0, ρ_0) with $\rho_0 > 0$ and x_0 being close, in the sense of (9.2.10), to the point $x^*(\rho_0)$.*

Consider the path-following method (cf. (9.2.11) - (9.2.12))

$$\rho_{t+1} = \left(1 + \frac{0.08}{\sqrt{\vartheta}}\right)\rho_t; \quad x_{t+1} = x_t - \frac{1}{1+\lambda(\Phi_{\rho_{t+1}}, x_t)}[\nabla_x^2 \Phi_{\rho_{t+1}}(x_t)]^{-1}\nabla_x \Phi_{\rho_{t+1}}(x_t). \quad (9.2.14)$$

Then the iterates of the method are well-defined, belong to the interior of G and the method possesses linear global rate of convergence:

$$f(x_t) - \min_G f \leq \frac{2\vartheta}{\rho_0}\left(1 + \frac{0.08}{\sqrt{\vartheta}}\right)^{-t}. \quad (9.2.15)$$

In particular, to make the residual in f less than a given $\epsilon > 0$, it suffices to perform no more that

$$N(\epsilon) \leq \lfloor 20\sqrt{\vartheta}\ln\left(1 + \frac{20\vartheta}{\rho_0\epsilon}\right)\rfloor \quad (9.2.16)$$

Newton steps.

We see that the parameter ϑ of the self-concordant barrier underlying the method is responsible for the *Newton complexity* of the method – the factor at the log-term in the complexity bound (9.2.16).

Remark 9.2.3 The presented result does not explain how to *start* tracing the path – how to get initial pair (x_0, ρ_0) close to the path. This turns out to be a minor difficulty: given in advance a strictly feasible solution $\bar{x}$ to (P), we could use the same path-following scheme (applied to certain artificial objective) to come close to the path $x^*(\cdot)$, thus arriving at a position from which we can start tracing the path. In our very brief outline of the topic, it makes no sense to go in these " details of initialization"; it suffices to say that the necessity to start from approaching $x^*(\cdot)$ basically does not violate the overall complexity of the method.

It makes sense if not to *prove* the aforementioned statements – the complete proofs, although rather simple, go beyond the scope of our lecture – but at least to *motivate* them – to explain what is the role of self-concordance and "magic inequality" (9.2.9) in ensuring properties **C.** and **D.** (this is all we need – the Theorem, of course, is an immediate consequence of these two properties).

Let us start with **C.** – this property is much more important. Thus, assume we are at a point x close, in the sense of (9.2.10), to $x^*(\rho)$. What this inequality actually says?

Let us denote by

$$\| h \|_{H^{-1}} = (h^T H^{-1} h)^{1/2}$$

the *scaled* Euclidean norm given by the inverse to the Hessian matrix

$$H \equiv \nabla_x^2 \Phi_\rho(x) = \nabla_x^2 F(x)$$

(the equality comes from the fact that Φ_ρ and F differ by a *linear* function $\rho f(x) \equiv \rho c^T x$). Note that by definition of $\lambda(\cdot,\cdot)$ one has

$$\lambda(\Phi_s, x) = \| \nabla_x \Phi_s(x) \|_{H^{-1}} \equiv \| sc + F'(x) \|_{H^{-1}}.$$

Due to the last formula, the closeness of x to $x^*(\rho)$ (see (9.2.10)) means exactly that

$$\| \nabla_x \Phi_\rho(x) \|_{H^{-1}} \equiv \| \rho c + F'(x) \|_{H^{-1}} \leq 0.05,$$

whence, by the triangle inequality,

$$\| \rho c \|_{H^{-1}} \leq 0.05 + \| F'(x) \|_{H^{-1}} \leq 0.05 + \sqrt{\vartheta} \tag{9.2.17}$$

(the concluding inequality here is given by (9.2.9) [8)], and this is the main point where this component of the definition of a self-concordant barrier comes into the play).

From the indicated relations

$$\lambda(\Phi_{\rho^+}, x) = \| \rho^+ c + F'(x) \|_{H^{-1}} \leq \| (\rho^+ - \rho) c \|_{H^{-1}} + \| \rho c + F'(x) \|_{H^{-1}} =$$

$$= \frac{|\rho^+ - \rho|}{\rho} \| \rho c \|_{H^{-1}} + \lambda(\Phi_\rho, x) \leq$$

[see (9.2.11), (9.2.17)]

$$\leq \frac{0.08}{\sqrt{\vartheta}}(0.05 + \sqrt{\vartheta}) + 0.05 \leq 0.134$$

(note that $\vartheta \geq 1$ by Definition 9.2.1). According to Proposition 6.2.2.(iii.3), Lecture 4, the indicated inequality says that *we are in the domain of quadratic convergence of the damped Newton method as applied to self-concordant function* Φ_{ρ^+}; namely, the indicated Proposition says that

$$\lambda(\Phi_{\rho^+}, x^+) \leq \frac{2(0.134)^2}{1 - 0.134} < 0.05.$$

as claimed in **C.**. Note that this reasoning heavily exploits self-concordance of F ■

To establish property **D.**, it requires to analyze in more details the notion of a self-concordant barrier, and we are not going to do it here. Just to demonstrate where ϑ comes from, let us prove an estimate similar to (9.2.13) for the particular case when, first, the barrier in question is the standard logarithmic barrier given by Example 9.2.1 and, second, the point x is exactly the point $x^*(\rho)$ rather than is close to the latter point. Under the outlined assumptions we have

$$x = x^*(\rho) \Rightarrow \nabla_x \Phi_\rho(x) = 0 \Rightarrow$$

[substitute expressions for Φ_ρ and F]

$$\rho c + \sum_{j=1}^{m} \frac{a_j}{b_j - a_j^T x} = 0 \Rightarrow$$

[take inner product with $x - x^*$, x^* being an optimal solution to (P)]

$$\rho c^T (x - x^*) = \sum_{j=1}^{m} \frac{a_j^T (x^* - x)}{b_j - a_j^T x} \leq$$

[take into account that $a_j^T (x^* - x) = a_j^T x^* - a_j^T x \leq b_j - a_j^T x$ due to $x^* \in G$]

$$\leq m,$$

whence

$$c^T (x - x^*) \leq \frac{m}{\rho} \equiv \frac{\vartheta}{\rho}$$

(for the case in question $\vartheta = m$). This estimate is twice better than (9.2.13) – this is because we have considered the case of $x = x^*(\rho)$ rather than the one of x close to $x^*(\rho)$.

[8)] indeed, for a positive definite symmetric matrix H it clearly is the same (why?) to say that $|g|_{H^{-1}} \leq \alpha$ and to say that $|g^T h| \leq \alpha |h|_H$ for all h

9.2.3.3 Applications

The most famous (although, we believe, not the most important) application of Theorem 9.2.4 deals with Linear Programming, when G is a polytope and F is the standard logarithmic barrier for this polytope (see Example 9.2.1). For this case, the Newton complexity of the method[9)] is $O(\sqrt{m})$, m being the # of linear inequalities involved into the description of G. Each Newton step costs, as it is easily seen, $O(mn^2)$ arithmetic operations, so that the *arithmetic cost per accuracy digit* – number of arithmetic operations required to reduce current inaccuracy by absolute constant factor – turns out to be $O(m^{1.5}n^2)$. Thus, we get a polynomial time solution method for LP, with complexity characteristics typically (for m and n of the same order) better than those for the Ellipsoid method (Lecture 7). Note also that with certain "smart" implementation of Linear Algebra, the above arithmetic cost can be reduced to $O(mn^2)$; this is the best known so far *cubic in the size of the problem* upper complexity bound for Linear Programming.

To increase list of application examples, note that our abilities to solve in the outlined style a convex program of a given structure are limited only by our abilities to point out self-concordant barrier for the corresponding feasible domain. *In principle*, there are no limits at all – it can be proved that *every* closed convex domain in $\mathbf{R}^n$ admits a self-concordant barrier with the value of parameter at most $O(n)$. This "universal barrier" is given by certain multivariate integral and is too complicated for actual computations; recall that we should form and solve Newton systems associated with our barrier, so that we need it to be "explicitly computable".

Thus, we come to the following important question:

How to construct "explicit" self-concordant barriers. There are many cases when we are clever enough to point out "explicitly computable self-concordant barriers" for convex domains we are interested in. We already know one example of this type – Linear Programming (although we do not know to the moment why the standard logarithmic barrier for a polytope given by m linear constraints is m-self-concordant; why it is so, it will become clear in a moment). What helps us to construct self-concordant barriers and to evaluate their parameters are the following extremely simple combination rules, completely similar to those for self-concordant functions (see Section 6.2.4.4, Lecture 4):

- [Linear combination with coefficients ≥ 1] Let F_i, $i = 1, ...m$, be ϑ_i-self-concordant barriers for the closed convex domains G_i, let the intersection G of these domains possess a nonempty interior Q, and let $\alpha_i \geq 1$, $i = 1, ..., m$, be given reals. Then the function
$$F(x) = \sum_{i=1}^{m} \alpha_i F_i(x)$$
is $(\sum_{i=1}^{m} \alpha_i \vartheta_i)$-self-concordant barrier for G.
- [Affine substitution] Let $F(x)$ be ϑ-self-concordant barrier for the closed convex domain $G \subset \mathbf{R}^n$, and let $x = A\xi + b$ be an affine mapping from $\mathbf{R}^k$ into $\mathbf{R}^n$ with the image intersecting int G. Then the composite function
$$F^+(\xi) = F(A\xi + b)$$
is ϑ-self-concordant barrier for the closed convex domain
$$G^+ = \{\xi \mid A\xi + b \in G\}$$

[9)] recall that it is the factor at the logarithmic term in (9.2.16), i.e., the # of Newton steps sufficient to reduce current inaccuracy by an absolute constant factor, say, by factor 2; cf. with the stories about polynomial time methods from Lecture 7

which is the inverse image of G under the affine mapping in question.

The indicated combination rules can be applied to the "raw materials" as follows:

- [Logarithm] The function
$$-\ln(x)$$
is 1-self-concordant barrier for the nonnegative ray $\mathbf{R}_+ = \{x \in \mathbf{R} \mid x > 0\}$;

[the indicated property of logarithm is given by 1-line computation]

- [Extension of the previous example: Logarithmic barrier, linear/quadratic case] Let
$$G = \mathrm{cl}\{x \in \mathbf{R}^n \mid \phi_j(x) < 0,\ j = 1, ..., m\}$$
be a nonempty set in $\mathbf{R}^n$ given by m convex quadratic (e.g., linear) inequalities satisfying the Slater condition. Then the function
$$f(x) = -\sum_{i=1}^{m} \ln(-\phi_i(x))$$
is m-self-concordant barrier for G.

[for the case when all the functions f_i are linear, the conclusion immediately follows from the Combination rules (a polyhedral set given by m linear inequalities is intersection of m half-spaces, and a half-space is inverse image of the nonnegative axis under affine mapping; applying the combination rules to the barrier $-\ln x$ for the nonnegative ray, we get, without any computations, that the standard logarithmic barrier is for a polyhedral set is m-self-concordant). For the case when there are also quadratic forms among f_i, you need a 1-2 lines of computations. Note that the case of *linear* f_i covers the entire Linear Programming, while the case of *convex quadratic* f_i covers much wider family of quadratically constrained convex quadratic problems.]

- The function
$$F(t, x) = -\ln(t^2 - x^T x)$$
is 2-self-concordant barrier for the "ice-cream cone"
$$\mathbf{K}^n_+ = \{(t, x) \in \mathbf{R} \times \mathbf{R}^n \mid t \geq |x|\};$$
the function
$$F(X) = -\ln \mathrm{Det}\ X$$
is n-self-concordant barrier for the cone $\mathbf{S}^n_+$ of positive definite symmetric $n \times n$ matrices.

One hardly could imagine how wide is the class of applications – from Combinatorial optimization to Structural Design and Stability Analysis/Synthesis in Control – of the latter two barriers, especially of the $\ln \mathrm{Det}$ -one.

9.2.3.4 Concluding remarks

The path-following scheme results in the most efficient, in terms of the worst-case complexity analysis, interior point methods for Linear and Convex Programming. From the practical viewpoint, anyhow, this scheme, in its aforementioned form, looks bad. The severe practical drawback of the scheme is its "short-step nature" – according to the scheme, the penalty parameter should be updated by the "programmed" rule (9.2.11), and this makes the actual performance of the method more or less close to the one given by (9.2.16). Thus, in the straightforward implementation of the scheme the complexity estimate (9.2.16) will be

not just the theoretical upper bound on the worst-case complexity of the method, but the indication of the "typical" performance of the algorithm. And a method *actually* working according to the complexity estimate (9.2.16) could be fine theoretically, but it definitely will be of very restricted practical interest in the large-scale case. For example, in LP program with $m \approx 10^5$ inequality constraints and $n \approx 10^4$ variables (these are respectable, but in no sense "outstanding" sizes for a practical LP program) estimate (9.2.16) predicts something like hundreds of Newton steps with Newton systems of the size $10^4 \times 10^4$; even in the case of good sparsity structure of the systems, such a computation would be much more time consuming than the one given by the Simplex Method.

In order to get "practical" path-following methods, we need a *long-step* tactics – rules for *on-line* adjusting the stepsizes in the penalty parameter to, let me say, local curvature of the path, rules which allow to update parameter as fast as possible – possible from the viewpoint of the actual "numerical circumstances" the method is in rather than from the viewpoint of very conservative theoretical worst-case complexity analysis.

Today there are efficient "long-step" policies of tracing the paths, policies which are both fine theoretically (i.e., satisfy complexity bound (9.2.16)) and very efficient computationally. Extremely surprising phenomenon here is that for "good" long-step path-following methods as applied to convex problems of the most important classes (Linear Programming, Quadratically Constrained Convex Quadratic Programming and some other) it turns out that

> *the actually observed number of Newton iterations required to solve the problem within reasonable accuracy is basically independent of the sizes of the problem and is within 30-50.*

This "empirical fact" (which can be only partly supported by theoretical considerations, not proved completely) is extremely important for applications; it makes polynomial time interior point methods the most attractive (and sometimes - the only appropriate) optimization tool in many important large-scale applications.

We should add that the efficient "long-step" implementations of the path-following scheme are relatively new, and for a long time the only interior point methods which demonstrated the outlined "data- and size-independent" convergence rate were the so called *potential reduction interior point methods*. In fact, the very first interior point method – the *method of Karmarkar* for LP – which initialized the entire interior point revolution, was a potential reduction algorithm, and what indeed caused the revolution was outstanding practical performance of this method. The method of Karmarkar possesses a very nice (and in fact very simple) geometry and is closely related to the interior penalty scheme; anyhow, time limitations enforce me to skip description of this wonderful, although now a little bit old-fashioned, algorithm.

The concluding remark we would like to make is as follows: all polynomial time implementations of the penalty/barrier scheme known so far are those of the barrier scheme (which is reflected in the name of these implementations: "interior point methods"); numerous attempts to do something similar with the penalty approach failed to be successful. It is a pity, due to some attractive properties of the scheme (e.g., here you do not meet with the problem of finding a feasible starting point, which, of course, is needed to start the barrier scheme).

Lecture 10

Augmented Lagrangians

Penalty methods studied in the previous lecture are very natural and general; unfortunately, they suffer from a serious drawback: to approximate well the solution to constrained problem, you have to work with large penalty parameters, and this inevitably makes the problem of unconstrained minimization of the penalized objective very ill-posed. The main advantage of the *augmented Lagrangian methods* is that they allow to approximate well the solution to a constrained problem by solutions of unconstrained (and penalized) auxiliary problems *without pushing the penalty parameter to infinity*; as a result, the auxiliary problems remain reasonably conditioned even when we are seeking for high-accuracy solutions. Let us look how all this works. We shall mainly focus on the case of equality constrained problem

$$\text{(ECP)} \quad f(x) \to \min \mid h(x) = \begin{pmatrix} h_1(x) \\ \dots \\ h_m(x) \end{pmatrix} = 0 \quad [x \in \mathbf{R}^n];$$

in the mean time I shall explain how the results for this case can be straightforwardly extended onto the general one.

10.1 Main ingredients

10.1.1 Local Lagrange Duality

Let x^* be a nondegenerate local solution to (ECP) (see Definition 4.2.1), let

$$L(x, \lambda) = f(x) + \lambda^T h(x) \equiv f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

be the Lagrange function of the problem, and let λ^* be the vector of Lagrange multipliers corresponding to the solution x^*, so that

$$\nabla_x L(x^*, \lambda^*) = 0, \tag{10.1.1}$$

and the matrix $\nabla_x^2 L(x^*, \lambda^*)$ is positive definite along the plane tangent at x^* to the feasible surface of the problem:

$$\nabla h(x^*) d = 0 \Rightarrow d^T \nabla_x^2 L(x^*, \lambda^*) d \geq \alpha |d|^2 \tag{10.1.2}$$

with certain positive $\alpha > 0$.

Assume for a moment that instead of (10.1.2) a stronger condition is satisfied:

(!) *the matrix $\nabla_x^2 L(x^*, \lambda^*)$ is positive definite on the entire space*

What would be the consequences of this stronger assumption?

The consequences are immediate: (10.1.1) together with (!) mean that x^* is a *nondegenerate local minimizer* of the Lagrange function $L(\cdot, \lambda^*)$. It follows that if

A. *We are clever enough to guess the vector λ^* of Lagrange multipliers,*
and
B. *We are lucky enough to be in the case (!),*

then we are be able to find x^* via unconstrained minimization applied to the function $L(\cdot, \lambda^*)$. Note that this is a fine smooth function which contains no penalty parameters and therefore is convenient for unconstrained minimization.

It remains to be clever and lucky to ensure **A** and **B**. Let us start with becoming clever, provided that *we already are lucky*, i.e., that *(!) indeed takes place*.

Those who have passed through the course Optimization I already know what is Lagrange duality: this is a machinery which allows to associate with constrained optimization problem (called *primal*) certain other optimization problem – the *dual* one; the solutions to the problems are closely related to each other. To enjoy all the advantages of this machinery, we should deal with *convex* primal problem, but something can be obtained also in the "locally convex case" (!).

Namely, one can prove the following

Theorem 10.1.1 *Under assumption (!) there exist*

- *convex neighbourhood V of the point x^* in $\mathbf{R}^n$,*
- *convex neighbourhood Λ of the point λ^* in $\mathbf{R}^m$,*

such that

(i) *The function*

$$L_\lambda(x) = L(x, \lambda)$$

for every $\lambda \in \Lambda$ is strongly convex in V and possesses uniquely defined critical point $x^*(\lambda)$ in V, the point being the nondegenerate minimizer of L_λ in V; the mapping $x^*(\lambda)$ is at least once continuously differentiable in Λ.

(ii) *The optimal value*

$$\underline{L}(\lambda) = L_\lambda(x^*(\lambda))$$

of the function $L_\lambda(x)$ in V is concave twice continuously differentiable function in Λ with the gradient

$$\nabla_\lambda \underline{L}(\lambda) = h(x^*(\lambda)). \tag{10.1.3}$$

The function $\underline{L}$ attains its maximum over $\lambda \in \Lambda$ at the point λ^*, the maximizer being unique and nondegenerate[1)].

The proof of the theorem is skipped.

Note that the Theorem says that under assumption (!) (x^*, λ^*) is *local saddle point* of the Lagrange function: there exists a neighbourhood of this point (namely, $V \times \Lambda$) such that L, being restricted onto the neighbourhood, attains at (x^*, λ^*) its minimum in x and its maximum in λ.

Let us look at the consequences of the Theorem; they are extremely encouraging. Indeed, we already know that if we were clever enough to guess what is λ^*, we could solve (ECP) by unconstrained minimization of $L_{\lambda^*}(x)$ – x^* is a nondegenerate unconstrained local minimizer of this function. And the Theorem says that there exists something which allows us to identify λ^* – this is the "locally dual" to (ECP) problem

$$\text{(D)} \quad \underline{L}(\lambda) \to \max;$$

this is fine problem of maximizing a smooth concave function, and λ^* is the unique (and nondegenerate) unconstrained maximizer of the function.

Of course, in order to solve (D), we should be able to compute the values and at least the first order derivatives of $\underline{L}$. Here again the above Theorem gives us all necessary tools: in order to compute the value of $\underline{L}(\lambda)$, we should minimize $L_\lambda(x)$ over V; according to item (i) of the Theorem, this latter

[1] of course, for maximization problem nondegeneracy of local solution means that the Hessian of the objective at the solution is negative definite

problem is well-posed and basically unconstrained: the Theorem says that for λ close enough to λ^* L_λ possesses unique critical point $x^*(\lambda)$ in V, and this point is nondegenerate local minimizer. Thus, we can approximate $x^*(\lambda)$ by unconstrained minimization tools. And from (10.1.3) it follows that $x^*(\lambda)$ gives us not only the value of $\underline{L}(\lambda)$ of the dual objective, but also its gradient – this gradient is simply the vector $h(\cdot)$ of constraints at $x^*(\lambda)$.

With these observations, we see that it is possible to approximate λ^* applying to $\underline{L}(\cdot)$ an unconstrained optimization method (Gradient Descent, Quasi-Newton or whatever else); in order to provide the method by the required first-order information at the current dual iterate λ, it suffices to find, again by unconstrained minimization, the minimizer $x^*(\lambda)$ of the function $L_\lambda(x)$. And what is nice in this picture, is that there are no "large parameters" – both the dual problem of maximizing $\underline{L}(\lambda)$ and the auxiliary primal problems of minimizing $L_\lambda(x)$ possess, at least in neighborhoods of their solutions, nice smoothness and nondegeneracy properties, and the values of the parameters responsible for these latter properties are not spoiled from iteration to iteration.

10.1.2 "Penalty cocknification"

The considerations of the previous subsection were conditional – "if we are lucky to be in situation (!), then...". What to do if we are not lucky to be in this situation? Well, what follows demonstrates that in order to become lucky it suffices to be smart – we always can enforce (!) to be the case, and what helps us to reach this goal, is the idea of penalization we know from the previous lecture. Namely, let us add to the actual objective f of (ECP) the quadratic penalty term, thus coming to the penalized objective

$$f_\rho(x) = f(x) + \frac{1}{2}\rho\sum_{i=1}^m h_i^2(x),$$

but *in contrast to what we did in the pure penalty method, let us keep the constraints as they were*, thus coming to the problem

$$(\mathrm{ECP}_\rho) \quad f_\rho(x) \to \min \mid h(x) = 0.$$

Since the penalty term is zero on the feasible surface, the new problem is equivalent to the initial one, and you can ask what is the goal of this strange manipulation. The answer is as follows:

(+) *if x^* is a nondegenerate solution to the original problem (ECP), then it is nondegenerate local solution to (ECP$_\rho$) as well* [which by itself is not that important] *and, moreover, for all large enough ρ this nondegenerate solution to (ECP$_\rho$) satisfies the crucial property (!).*

The immediate consequence of (+) is as follows: passing from (ECP) to equivalent problem (ECP$_\rho$), we may ensure (!) and thus get all nice possibilities of solving (ECP$_\rho$) (or, which is the same, (ECP)) outlined in the previous section.

Even if you believe in (+), you may ask: all this is fine, but we again meet with "large enough" ρ; this large ρ will finally occur in the objectives $L_\lambda(\cdot)$ of the auxiliary primal problems given by the scheme of the previous section, and we know from our discussion of the penalty methods that it would make these auxiliary problems very difficult for numerical solution. This could be a quite reasonable objection, if there were no crucial difference between the situation with penalty methods and our now situation. In the penalty scheme, the value of the penalty parameter is directly linked with the accuracy ϵ to which we are going to solve the problem: as we remember, violations of the constraints at the unconstrained minimizer of penalized objective are of order of $1/\rho$, and to make these violations $\le \epsilon$ we *must* work with penalty parameter of order of $1/\epsilon$. The role of penalization now is quite different: it should enforce (!), i.e., positive definiteness (on the entire space) of the Hessian of the Lagrangian $L^\rho(x,\lambda)$ of the penalized problem, and (+) says that all large enough values of ρ are appropriate for this purpose. Whenever a given ρ results in positive definite $\nabla_x^2 L^\rho(x^*,\lambda^*)$, we may use this value of the penalty in the scheme of the previous section to solve problem (ECP$_\rho$) (or, which is the same, our original problem – they are equivalent!) to arbitrarily high accuracy ϵ. Thus, in our now situation we are not enforced to push the penalty parameter to infinity and will be satisfied by fixed value of it; whether this value indeed must be large and will cause numerical difficulties when solving auxiliary primal problems, or it is quite moderate

and no problems of this type occur, it depends on local properties of the data, but not on the accuracy to which the problem should be solved. The simplest way to see the indicated difference is to look what happens in the convex case (linear constraints, convex objective). It can be easily seen that here *each* positive value of ρ makes $\nabla^2 L^\rho(x^*, \lambda^*)$ positive definite, so that each value of ρ is "large enough"; note that in the penalty scheme we should work with indeed large penalties independently of whether the problem is convex or not.

It is time now to support our claim (+). To see that (+) indeed is valid, let us denote by λ^* the vector of Lagrange multipliers associated with x^* in the original problem (ECP); as we know, this vector is uniquely defined by the KKT condition

$$\nabla_x f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla h_i(x^*) = 0.$$

Now, the gradient of the penalized objective is

$$\nabla_x f_\rho(x) = \nabla_x f(x) + \rho \sum_{i=1}^m h_i(x) \nabla h_i(x),$$

and we see that at the feasible surface (and, in particular, at the point x^*), where $h(x) = 0$, the gradients of the actual and the penalized objectives coincide with each other; thus, λ^* serves as the vector of Lagrange multipliers associated with x^* in problem (ECP$_\rho$) as well.

Now let us compute the Hessian with respect to x of the Lagrange function

$$L^\rho(x, \lambda) = f_\rho(x) + \sum_{i=1}^m \lambda_i h_i(x) \equiv f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \frac{1}{2}\rho \sum_{i=1}^m h_i^2(x) \equiv$$

$$\equiv L(x, \lambda) + \frac{1}{2}\rho \sum_{i=1}^m h_i^2(x) \tag{10.1.4}$$

of the modified problem (ECP$_\rho$) (from now on, $L(x, \lambda)$ means the Lagrangian of the original problem). We have

$$\nabla_x L^\rho(x, \lambda) = \nabla_x L(x, \lambda) + \rho \sum_{i=1}^m h_i(x) \nabla_x h_i(x),$$

whence

$$\nabla_x^2 L^\rho(x, \lambda) = \nabla_x^2 L(x, \lambda) + \rho \sum_{i=1}^m \nabla_x h_i(x) [\nabla_x h_i(x)]^T + \rho \sum_{i=1}^m h_i(x) \nabla_x^2 h_i(x).$$

Introducing notation

$$H(x, \lambda) = \nabla_x^2 L(x, \lambda), \quad A(x) = \nabla_x h(x) \equiv \begin{pmatrix} [\nabla h_1(x)]^T \\ \dots \\ [\nabla h_m(x)]^T \end{pmatrix}, \quad R(x) = \sum_{i=1}^m h_i(x) \nabla_x^2 h_i(x),$$

we can rewrite the resulting equality as

$$\nabla_x^2 L^\rho(x, \lambda) = H(x, \lambda) + \rho A^T(x) A(x) + \rho B(x). \tag{10.1.5}$$

We would like to prove that the left hand side in this relation at the point $(x = x^*, \lambda = \lambda^*)$ is positive definite, provided that ρ is large enough. Now let us make the following useful observations:

- $H(x^*, \lambda^*)$ is positive definite at the subspace T of directions tangent to the feasible surface at the point x^*; this is the second order part of the Second Order Sufficient Optimality condition which is assumed to be satisfied at x^* (recall that we all the time are speaking about the case when x^* is a nondegenerate solution to (ECP)). As we know, T is exactly the kernel of the matrix $\nabla h(x^*) = A(x^*)$:

$$T = \{d \mid A(x^*) d = 0\};$$

- the matrix $A^T(x^*)A(x^*)$ is positive semidefinite (as any matrix of the form B^TB: indeed, $d^TB^TBd = |Bd|^2$); its kernel, as it is immediately seen from the computation in parentheses, is exactly the kernel of $A(x^*)$, i.e., T;
- The matrix $R(\cdot)$ simply vanishes at x^*, due to presence of factors $h_i(x)$ in the expression for the matrix; at the point x^* – this point is feasible for (ECP) – these factors become zero, so that $R(x^*) = 0$.

We conclude that

$$\nabla_x^2 L^\rho(x^*, \lambda^*) = H(x^*, \lambda^*) + \rho A^T(x^*)A(x^*),$$

and the fact that the right hand side is positive definite for large enough ρ is an immediate consequence of the following simple fact (prove it!):

Lemma 10.1.1 *Let D be positive semidefinite matrix and C be a symmetric matrix which is positive definite on the kernel of D. Then the matrix $C + \rho D$ is positive definite for all large enough values of ρ.*

To derive (+) from above observations, it suffices to apply the Lemma to $D = A^T(x^*)A(x^*)$ and $C = H(x^*, \lambda)$.

10.2 Putting things together: Augmented Lagrangian Scheme

The algorithmic scheme implementing the above ideas is called the Augmented Lagrangian method, or the Multipliers method; the first name comes from the fact that the Lagrange function (10.1.4) of the problem (ECP_ρ) can be treated as certain "augmentation" of the Lagrange function of the original problem (ECP); the second name stresses the leading role of the Lagrange multipliers in the method.

The generic Augmented Lagrangian scheme is as follows. Assume that we already have chosen somehow value of the penalty parameter ρ and thus deal with problem

$$(\mathrm{ECP}_\rho) \quad f(x) + \frac{\rho}{2}|h(x)|^2 \to \min \mid \; h(x) = 0$$

equivalent to the original problem (ECP), and let ρ be large enough to ensure (!) – let the Hessian $\nabla_x^2 L^\rho(x, \lambda)$ of the Lagrange function

$$L^\rho(x, \lambda) = f(x) + \frac{\rho}{2}\sum_{i=1}^m h_i^2(x) + \sum_{i=1}^m \lambda_i h_i(x)$$

taken at the point (x^*, λ^*) be positive definite on the entire space; here x^* is the nondegenerate solution of (ECP) to be approximated and λ^* is the corresponding vector of Lagrange multipliers. To make further considerations clear, let us summarize what is said about the situation by the discussion of Section 10.1.1.

We know that

- to find λ^* is the same as to solve the dual problem

$$(\mathrm{D}^\rho) \quad \underline{L}^\rho(\lambda) \equiv \min_x L^\rho(x, \lambda) \to \max$$

($\min_x$ here should be taken locally, as was explained in Section 10.1.1);

- the objective in the dual problem is everywhere concave and twice continuously differentiable in a neighbourhood of λ^*, and λ^* is nondegenerate solution to the dual problem; thus, the dual problem is well-suited for numerical solution, *provided that we are able to compute the value and the gradient of the dual objective at a point*
- we do have a possibility to compute the value and the gradient of the dual objective at a point, at least at a point close to λ^*: to this end it suffices to find the solution $x^\rho(\lambda)$ of the auxiliary unconstrained problem

$$(\mathrm{P}_\lambda^\rho) \quad \min_x L^\rho(x, \lambda)$$

(where $\min_x$ again should be taken locally) and to set

$$\underline{L}^\rho(\lambda) = L^\rho(x^\rho(\lambda), \lambda), \quad \nabla_\lambda \underline{L}^\rho(\lambda) = h(x^\rho(\lambda)). \tag{10.2.1}$$

Now we may run any method for unconstrained maximization of the dual objective $\underline{L}^\rho$ (*"outer iterations"* generating sequence $\lambda_0, \lambda_1, ...$ of iterates converging to λ^*), accompanying every step t of the method by solving the auxiliary problem $(\mathrm{P}^\rho_{\lambda_{t-1}})$ in order to find

$$x_{t-1} = x^\rho(\lambda_{t-1})$$

and thus to get the first order information on the dual objective which is required to update λ_{t-1} into λ_t.

Implementations of the scheme differ from each other mainly by methods we use to maximize the dual objective and to solve the auxiliary primal problems.

10.2.1 Solving auxiliary primal problems $(\mathrm{P}^\rho_\lambda)$

The best choice here is, of course, the Newton method with linearizes or the modified Newton method, when the second order information on the objective and the constraints is available. If it is not the case, we could use Quasi-Newton methods, Conjugate Gradient, etc. When solving the auxiliary problems, we should use reasonable *safeguard policy* in order to ensure that our minimization is indeed local. Since we do not know in advance what is the neighbourhood V introduced in Section 10.1.1, this is not an easy task. The simplest policy here is to use the solution to the previous auxiliary problem as the starting point when solving the current one and to take specific care of linearizes (e.g., to move until the closest minimum of the current auxiliary objective on the current search line). With reasonable tactics of this type we indeed may ensure that if the starting point in the very first auxiliary problem is close enough to x^* and the initial guess for λ^* is close enough to λ^*, then at all steps of the method we shall indeed approximate $x^\rho(\cdot)$ and not other tentative local solutions to the auxiliary primal problems. And of course to use the previous primal iterate as the starting point in the new auxiliary primal problem saves a lot of computational effort.

10.2.2 Solving the dual problem

When solving the dual problem (D^ρ) – this is the upper-level process in our two-level scheme – we are less flexible when choosing the optimization method: since the information on the dual objective is "implicit" – to get its value and gradient, we should solve an auxiliary primal optimization problem, it normally is computationally expensive, and things like line search become very costly. The simplest algorithm we could use in the outer iterates is the following scheme:

$$\lambda_t = \lambda_{t-1} + \rho h(x_{t-1}). \tag{10.2.2}$$

Relation (10.2.2) has a very simple motivation: since x_{t-1} is unconstrained minimizer of $L^\rho(\cdot, \lambda_{t-1})$, we have

$$0 = \nabla_x L^\rho(x_{t-1}, \lambda_{t-1}) = \nabla_x f(x) + \sum_{i=1}^m \rho h_i(x) \nabla_x h_i(x) + \sum_{i=1}^m (\lambda_{t-1})_i \nabla h_i(x) =$$

$$= \nabla_x f(x_{t-1}) + \sum_{i=1}^m [(\lambda_{t-1})_i + \rho h_i(x_{t-1})] \nabla h_i(x).$$

Comparing this equality with the one

$$0 = \nabla_x f(x^*) + \sum_{i=1}^m \lambda^*_i \nabla_x h_i(x^*)$$

defining the Lagrange multipliers λ^*, we see that the right hand side in (10.2.2) can be viewed as a natural approximation to λ^*, the better the closer is x_{t-1} to x^*.

This is only motivation, of course; now let us look what indeed can be said about recurrence (10.2.2). From (10.2.1) we observe that $h(x_{t-1})$ is exactly the gradient of the dual objective at λ_{t-1}, so that (10.2.2) is in fact Gradient Ascent method for the dual problem with constant stepsize ρ (recall that the dual problem is the maximization one, so that a gradient-type method here should perform steps along the gradient rather than along the intergradient, as it is in the minimization case; this is why we are speaking about Gradient *Ascent*).

10.2.2.1 Dual rate of convergence

To understand whether the stepsize ρ in Gradient Ascent is or is not good for us, let us look at the eigenvalue structure of the Hessian of the dual objective at λ^* – we know from our investigation of the Gradient method that this structure is exactly what governs the asymptotical rate of convergence of the method.

The Hessian of the dual objective can be found as follows (to save notation, I write $x(\lambda)$ instead of $x^\rho(\lambda)$): we already know that

$$\nabla_\lambda \underline{L}^\rho(\lambda) = h(x(\lambda))$$

and $x(\cdot)$ is continuously differentiable in a neighbourhood of $\lambda = \lambda^*$ by Theorem 10.1.1. Differentiating both sides in λ, we get

$$\nabla^2_\lambda \underline{L}^\rho(\lambda) = \nabla_x h(x(\lambda))\nabla_\lambda x(\lambda). \tag{10.2.3}$$

To find $\nabla_\lambda x(\lambda)$, let us differentiate in λ the relation

$$\nabla_x L^\rho(x(\lambda), \lambda) = 0,$$

which defines $x(\lambda)$. We get

$$\nabla^2_x L^\rho(x(\lambda), \lambda)\nabla_\lambda x(\lambda) + \frac{\partial^2}{\partial\lambda\partial x} L^\rho(x(\lambda), \lambda) = 0;$$

as it is immediately seen from the expression for L^ρ, the second term in the left hand side of the latter equation is $[\nabla_x h(x(\lambda))]^T$, and we get

$$\nabla_\lambda x(\lambda) = -[\Phi^\rho(\lambda)]^{-1}[\nabla_x h(x(\lambda)]^T,$$

Φ^ρ being the Hessian of the augmented Lagrangian L^ρ with respect to x taken at the point $(x(\lambda), \lambda)$. Substituting this expression into (10.2.3), we get

$$\nabla^2_\lambda \underline{L}^\rho(\lambda) = -[\nabla_x h(x(\lambda))][\Phi^\rho(\lambda)]^{-1}[\nabla_x h(x(\lambda)]^T, \tag{10.2.4}$$

Substituting expression for the Hessian of the augmented Lagrangian L^ρ in x (see (10.1.5), we get the following important conclusion:

The Hessian Ψ^ρ of the dual objective $\underline{L}^\rho(\cdot)$ at the point λ^* is given by

$$\Psi^\rho = -A[H + \rho A^T A]^{-1} A^T, \tag{10.2.5}$$

$A = \nabla_x h(x^*)$ and

$$H = \nabla^2_x f(x^*) + \sum_{i=1}^m \lambda_i \nabla^2_x h_i(x^*)$$

is the Hessian with respect to x of the Lagrange function of the original problem (ECP), the Hessian being taken at the point (x^*, λ^*).

Relation (10.2.5) allows to make the following crucial observation:

Proposition 10.2.1 *The matrix $-\rho\Phi^\rho$ tends to the unit matrix I as $\rho \to \infty$.*

Proof. Assuming, for the sake of simplicity, that the matrices H and $AH^{-1}A^T$ are nonsingular (this assumption can be easily eliminated by a little bit more complicated reasoning) and applying Superman-Morrison formula, we get

$$[H+\rho A^TA]^{-1} = H^{-1} - H^{-1}A^T[\rho^{-1}I_m + AH^{-1}A^T]^{-1}AH^{-1};$$

denoting $Q = AH^{-1}A^T$ and taking into account (10.2.5), we get

$$-\Psi^\rho = Q - Q[\rho^{-1}I_m + Q]^{-1}Q.$$

Applying the formula[2)]

$$[Q+\rho^{-1}I_m]^{-1} = \left[Q(I_m+\rho^{-1}Q^{-1})\right]^{-1} = [I_m+\rho^{-1}Q^{-1}]^{-1}Q^{-1} =$$

$$= [I_m - \rho^{-1}Q^{-1} + o(\rho^{-1})]Q^{-1} = Q^{-1} - \rho^{-1}Q^{-2} + o(\rho^{-1}),$$

we come to

$$-\Psi^\rho = Q - Q[Q^{-1} - \rho^{-1}Q^{-2} + o(\rho^{-1})]Q = \rho^{-1}I_m + o(\rho^{-1}),$$

and the conclusion follows. ∎

The result stated in the Proposition is indeed crucial for us, due to the following

Corollary 10.2.1 [Dual convergence rate] *Let ρ be large enough, and let the starting point λ_0 in recurrence (10.2.2) be close enough to λ^*. Then recurrence (10.2.2) converges to λ^* linearly with the convergence ratio $\kappa(\rho)$ which tends to 0 as $\rho \to \infty$.*

The statement given by the corollary is indeed very important: it establishes (local) linear convergence of simple and easily implementable recurrence (10.2.2), provided that ρ is large enough; moreover, the convergence can be done arbitrarily fast, if ρ is large enough. Of course, this observation does not say that in actual computations you should set ρ to a huge value and get the solution in one step: this "one step" is one step of outer iteration in λ, and to perform this step, you should solve the auxiliary primal problem with huge value of penalty, which, as we already know, is very difficult (note also that we know without any dual considerations that solving the auxiliary primal problem with huge ρ we get good approximate solution to (ECP) – this is the penalty method). The actual meaning of the statement is, of course, that we may hope to deal with "large enough" value of the penalty which will ensure reasonable rate of convergence of (10.2.2) and at the same time will be not "too large" to make the auxiliary primal problems very difficult.

It is time now to prove the Corollary. I shall give the model of the proof only. Namely, let us believe that the behaviour of recurrence (10.2.2) started close enough to λ^* is basically the same as if the dual objective was the quadratic function

$$\psi(\lambda) = \frac{1}{2}(\lambda-\lambda^*)^T\Psi^\rho(\lambda-\lambda^*);$$

it indeed is so, but let us skip the required dull (although simple) justification. With our quadratic model of the dual objective, the result can be obtained in one line. Recall that (10.2.2) is Gradient Ascent with stepsize ρ, so that

$$\lambda_t - \lambda^* = \lambda_{t-1} - \lambda^* + \rho\nabla_\lambda\psi(\lambda_{t-1}) = (I_m + \rho\Psi^\rho)(\lambda_{t-1}-\lambda^*);$$

thus, the residual vector $\lambda_t - \lambda^*$ is multiplied at each step by the matrix $I_m + \rho\Psi^\rho$, and this matrix tends to 0 as $\rho \to \infty$ by Proposition 10.2.1 and therefore is close to 0 for large ρ. ∎

[2)] coming from the standard expansion

$$(I_m + \epsilon S)^{-1} = I_m - \epsilon S + \epsilon^2 S^2 - \epsilon^3 S^3 + ...$$

which is valid, for all small in absolute value ϵ, also in the matrix case

One may ask whether indeed simple recurrence (10.2.2) is the best we can use. The question is quite reasonable: for fixed ρ – and our main motivation of the entire method was the desire to deal with perhaps large, but fixed value of the penalty parameter – the rate of convergence of this recurrence is linear. It is easily seen that the distance between x_t and the exact solution x^* is of order of distance between λ_t and λ^*, so that with recurrence (10.2.2) we get only linear convergence of both primal and dual variables. Could we get something better? Of course we could: relation (10.2.4) expresses the Hessian of the dual objective via the data which are available after $x(\lambda) = x^\rho(\lambda)$ is found from the solution of the auxiliary primal problem; thus, we could solve the dual problem by the Newton method, thus ensuring quadratic rate of local convergence of the scheme.

10.2.3 Adjusting penalty parameter

There is one more important issue to be discussed: how to choose ρ? For the time being, we simply have assumed that the penalty parameter is chosen in a way which ensures (!). What we know is that *all* large enough values of the penalty fit this requirement, but how to choose an appropriate ρ in practice? Advice like "set ρ to something huge" definitely is of no interest: with this approach, we convert the method into the penalty one and loose all computational advantages associated with possibility to deal with moderate values of the penalty.

The simplest way to *tune* the penalty is as follows. As we know from the previous analysis, when ρ is "appropriately large" to ensure linear rate of convergence of (10.2.2) with reasonable convergence ratio, then the norm of the gradient of the dual objective also should decrease linearly with reasonable convergence ratio. We know what is this gradient – this is the vector of constraints at the solution to the corresponding auxiliary primal problem; thus, for appropriately large ρ the quantities $|h(x_t)|$ decrease with t in geometric progression with the ratio which, in principle, can be done arbitrarily close to 0, provided that ρ is large enough. Let us use this observation as the driving force in our policy for adjusting the penalty. Namely, let us start the procedure with certain ad hoc value of ρ and look whether the residuals $|h(x_t)|$ indeed reasonably decrease with t, say, at least by factor 0.25 each time. Until it is so, we keep the penalty constant; and if at certain step t the current value of $|h(x_t)|$ is too large – exceeds $0.25|h(x_{t-1}|$ – we interpret this as a signal to increase the penalty by certain factor, say, 10. Then we resolve the auxiliary primal problem with this new value of the penalty and the same as before value of λ, again look whether we got required progress in constraint violation, and if not – again increase the penalty, and so on.

Of course, we could use another "driving forces" in penalty updating policy, but the progress in constraint violation is the most natural one: all we are interested in is in fact to make the constraint violation small. Indeed, at the solution x_t to the auxiliary primal problem the gradient of the objective for sure is linear combination of the gradients of constraints (write down the Fermat rule for the auxiliary problem), so that the violations of the constraints form the only obstacle for x_t to be KKT point of (ECP); thus, it is very natural to update the penalty on the basis of progress in diminishing this obstacle.

Let me also note that the indicated policy in fact incorporates both the penalty and the Lagrange Multiplier technique. Indeed, this policy increases the penalty until the solution to the current auxiliary primal problem results in prescribed progress in constraint violation; independently of what we do with the Lagrange multipliers and how high is the rate of dual convergence, we will achieve eventually this goal via the penalty mechanism.

10.3 Incorporating Inequality Constraints

For the moment, we know only how to apply the Augmented Lagrangian Scheme to an equality constrained problem. In fact there is no difficulty to incorporate into the Augmented Lagrangian Scheme inequality constraints, and to this end it suffices to use the reduction of a general constrained problem to an equality constrained one (Lecture 8). Namely, given a constrained problem

$$\text{(GCP)} \quad \min_x \{f(x) : h_i = 0,\ i = 1, ..., m,\ g_j(x) \le 0,\ j = 1, ..., k\},$$

we can transform it equivalently into the equality constrained problem

$$(\mathrm{ECP}^*) \quad \min_x \left\{ f(x) : h_i(x) = 0,\ i = 1, ..., m,\ g_j(x) + s_j^2 = 0,\ j = 1, ..., k \right\}$$

with extended list of variables. Applying the Augmented Lagrangian Scheme to the resulting problem, we come to the necessity to work with the augmented Lagrangian

$$L^\rho(x, s; \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^k \mu_j [g_j(x) + s_j^2] + \frac{\rho}{2} \left[\sum_{i=1}^m h_i^2(x) + \sum_{j=1}^k [g_j(x) + s_j^2]^2 \right];$$

what we should do with this Lagrangian is to find its (local) saddle point, and to this end the scheme in question suggests to maximize with respect to the dual variables λ, μ the function

$$\underline{L}^\rho(\lambda, \mu) = \min_{x,s} L^\rho(x, s; \mu, \lambda).$$

In fact we have no problems with minimization with respect to s_j in the right hand side of the latter relation. Indeed, explicit computation results in

$$L^\rho(x, s; \lambda, \mu) = f(x) + \frac{\rho}{2} \sum_{j=1}^k \left[g_j(x) + s_j^2 + \frac{\mu_j}{\rho} \right]^2 + \sum_{i=1}^m \lambda_i h_i(x) + \frac{\rho}{2} \sum_{i=1}^m h_i^2(x) - \sum_{j=1}^k \frac{\mu_j^2}{2\rho}.$$

It follows, consequently, that minimization of the Lagrange function with respect to the part s of primal variables results in setting s_j^2 equal to the quantity $-(g_j(x) + \mu_j/\rho)$, when this quantity is nonnegative, or to zero, when this quantity is negative. Consequently,

$$\underline{L}^\rho(\lambda, \mu) = \min_x \{ f(x) + \frac{\rho}{2} \sum_{j=1}^k \left(g_j(x) + \frac{\mu_j}{\rho} \right)_+^2 + \sum_{i=1}^m \lambda_i h_i(x) + \frac{\rho}{2} \sum_{i=1}^m h_i(x)^2 \} - \sum_{j=1}^k \frac{\mu_j^2}{2\rho}, \qquad (10.3.1)$$

where, as in the previous Lecture, $(a)_+$ is the positive part of real a (a itself, if it is positive, and 0 otherwise).

Thus, the auxiliary primal problems arising in the Augmented Lagrangian Scheme in fact are problems in the initial design variables.

From theoretical viewpoint, the following question is important. Validity of the Augmented Lagrangian Scheme for equality constrained problem (i.e., the theoretical local convergence results of the previous sections) required nondegeneracy of the solution x^* to be approximated. Thus, we should understand when a locally optimal solution x^* to (GCP), extended in the evident way (by setting $s_j^* = \sqrt{-g_j(x^*)}$) to solution (x^*, s^*) of (ECP*), results in a nondegenerate solution to the latter problem.

The answer is as it should be:

Proposition 10.3.1 *If x^* is a nondegenerate solution to* (GCP) [3], *then (x^*, s^*) is nondegenerate solution to* (ECP*).

The (simple) proof of this fact is left to the reader.

10.4 Convex case: Augmented Lagrangians; Decomposition

All results of Section 10.1.1 were local; this is quite natural, since there we dealt with general-type nonconvex program. In the case of *convex* program all results can be globalized; let us briefly outline the good things happening in this case. Thus, consider a *convex* optimization program

$$(\mathrm{CnvP}) \quad \min_x \{ f(x) : g_j(x) \le 0,\ j = 1, ..., k \},$$

[3] Definition 4.2.1: x^* is regular for the constraints, satisfies the Second Order Sufficient Optimality conditions and, besides this, Lagrange multipliers related to all inequality constraints active at x^* are strictly positive – this is called *strict* complementary slackness

where f and g_j are convex and twice continuously differentiable functions on $\mathbf{R}^n$. From now on, let us assume that the problem is *solvable* and *satisfies the Slater condition*: there exists $\bar{x}$ such that all constraints are strictly negative at $\bar{x}$.

Those who have passed through the course "Optimization I" know, and other should believe that under the indicated assumptions one can define the problem *dual* to (Rsvp), namely, the problem

$$\text{(CnvD)} \quad \max_{\mu} \{\phi^*(\mu) : \mu \geq 0\},$$

where $\mu = (\mu_1, ..., \mu_k)$ are the *dual variables* and

$$\phi^*(\mu) = \inf_{x \in \mathbf{R}^n} L(x, \mu),$$

with

$$L(x, \mu) = f(x) + \sum_{j=1}^{k} \mu_j g_j(x)$$

being the classical Lagrange function of problem (Rsvp). From the construction of the dual problem it follows that the dual objective is concave function taking values in the extended by $-\infty$ real axis. Due to the concavity of the dual objective, (CnvD) is in fact *convex* optimization problem (this is the same – to maximize a concave function ϕ^* or to minimize its negation $-\phi^*$, which is a convex function).

The links between the primal problem (Rsvp) and its dual (CnvD) are given by the *Convex Programming Duality Theorem* which states that

If (Rsvp) *is solvable and satisfies the Slater condition, then* (CnvD) *also is solvable, and the optimal values in the problems are equal to each other. In this case a feasible solution x^* to* (Rsvp) *and a feasible solution μ^* to* (CnvD) *are optimal for the corresponding problems if and only if (x^*, μ^*) is a saddle point of the Lagrange function $L(x, \mu)$ on $\mathbf{R}^n \times \mathbf{R}^k_+$ (i.e., $L(x, \mu^*)$ attains its minimum over $x \in \mathbf{R}^n$ at $x = x^*$, while $L(x^*, \mu)$ attains its maximum over $\mu \geq 0$ at $\mu = \mu^*$).*

In particular, all optimal solutions to (Rsvp) *are among unconstrained minimizers of the function $L(x, \mu^*)$, μ^* being any optimal solution of the dual problem.*

Note that Theorem 10.1.1 is a local version of this Duality Theorem.

The Duality Theorem has a lot of applications, both of theoretical and of computational type. From the *computational* viewpoint, there are two ways to exploit the theorem:

- There are cases when the structure of the data in (Rsvp) allows to compute the dual objective analytically (this is the case in Linear, Linearly Constrained Quadratic and Geometric Programming). Whenever this is the case, we may gain a lot solving (CnvD) instead of (Rsvp) and then recovering the primal solutions from the dual ones (which sometimes is easy and sometimes is not, see below).

Computational consequences of this type clearly are restricted by our abilities to compute the dual objective analytically; in our today lecture we are not interested in these (in fact very important) "particular cases".

- In the general case we can implement the Lagrange Multipliers scheme, namely, to solve (CnvD) numerically by a first-order method for convex optimization under *linear* inequality constraints (note that the only explicit inequalities in (CnvD) are $\mu_j \geq 0$); thus, we may switch from *nonlinearly* constrained primal problem to a problem with *simple linear* inequalities. Of course, to solve the dual problem, we need possibility to compute its objective and its gradient at a given point; same as in the Augmented Lagrangian Scheme for general-type problems, to this end we can solve numerically the problem of minimizing the Lagrangian in x with $\mu \geq 0$ being fixed. In our now situation, this "auxiliary primal problem" is fine - with smooth convex objective, so that we have good possibilities to approximate the global solution to the auxiliary problem.

10.4.1 Augmented Lagrangians

In this section we focus on the second of the aforementioned issues, on possibility to reduce a general type primal convex program to a linearly constrained dual one.

The outlined scheme is "globalization" of the one developed in the previous sections, but, for the time being, without "augmenting" the Lagrangian. What is fine is that now we are in convex situation, so that all "local minimizations" from Section 10.1.1 can be replaced by global ones, we should not bother much about "good enough" primal and dual starting points, safeguards, etc.

There is, anyhow, a weak point in the classical Lagrange function. Namely, it may happen that

- the dual objective ϕ^* may be nonsmooth and even infinite at some points; whenever it is the case, it is unclear how to solve the dual problem numerically

The indicated difficulty is not something rare: this is exactly what happens in the Linear Programming case. Here the dual objective turns out to be linear in certain polytope and to be $-\infty$ outside this polytope, and (CnvD) (the usual LP dual program) normally is as difficult as the primal problem.

- it may happen that the auxiliary primal problem *associated with the optimal vector μ^* of Lagrange multipliers*, the one being the optimal solution to (CnvD), has many optimal solutions. The Convex Programming Duality Theorem says that all optimal solutions to (Rsvp) for sure are optimal for the auxiliary primal problem in question, but the Theorem does not say that the latter problem has no other optimal solutions. Whenever it has "extra" optimal solutions, we meet with nontrivial problem how to select the solutions we actually are interested in, i.e, those optimal for (Rsvp), from the minimizers of $L(\cdot, \mu^*)$.

This second difficulty also is not something rare: in the case of Linear Programming the function $L(x, \mu^*)$, μ^* being optimal solution to (CnvD), is constant, so that its optimal set – the entire $\mathbf{R}^n$ – says nothing about the actual optimal set of (Rsvp).

In fact both indicated difficulties have the same origin: possible non-existence or non-uniqueness of optimal solution to the auxiliary problem

$$\min_x L(x, \mu) \tag{10.4.1}$$

If we somehow assure existence and uniqueness of the solution $x^*(\lambda)$ in the latter problem and its continuity in μ, then we, first, will get continuously differentiable dual objective. Indeed, it can be proved (cf. Section 10.1.1) that in the case in question

$$\nabla_\mu \phi^*(\mu) = g(x^*(\mu)), \tag{10.4.2}$$

which is the continuous function of μ, provided that $x^*(\mu)$ is continuous. Second, in the case under consideration we clearly shall be able to restore the optimal solution to the primal problem via the optimal solution to the dual.

There is one evident case when problem (10.4.1) has exactly one solution: this is the case when the sum

$$r(x) = f(x) + \sum_{j=1}^{k} g_j(x)$$

of the objective and the constraints is *strongly convex* (with positive definite Hessian at every point) and grows faster than $|x|$ as $|x| \to \infty$:

$$r(x)/|x| \to \infty, \ \ |x| \to \infty.$$

Indeed, in this case it is easily seen that the objective in (10.4.1) for every $\mu > 0$ also is strongly convex and goes to infinity as $|x| \to \infty$; consequently, (10.4.1) has unique solution (which, as it is easily seen, is continuous in $\mu > 0$). Consequently, in the case under consideration application of the Lagrange Multipliers Scheme causes no conceptual difficulties.

What to do if the just mentioned sufficient condition of "well-posedness" of (10.4.1) is not satisfied? A good idea is to pass to *augmented Lagrange function* – basically the same idea we used in the general case

of nonconvex problems. Namely, let θ_j be *increasing strongly convex* functions on the axis normalized by the conditions

$$\theta_j(0) = 0, \;\; \theta_j'(0) = 1.$$

The problem

$$(\text{CnvP}^*) \quad \min_x \{f(x) : \theta_j(g_j(x)) \le 0, \; j = 1, ..., k\}$$

clearly is equivalent to (Rsvp) and, as it is easily seen, also is convex. One can straightforwardly verify that, due to the normalization conditions on θ_j, the vector of optimal Lagrange multipliers for the new problem is the same as for the old one (the new dual objective is, anyhow, different from the old one). Under mild regularity assumptions one can prove that the modified problem satisfies the aforementioned sufficient condition for well-posedness of problems (10.4.1); e.g., if all g_j are linear, this sufficient condition is satisfied whenever the feasible domain of (Rsvp) is bounded. Thus, by passing from (Rsvp) to equivalent problem (Rsvp*), or, which is the same, passing from the classical Lagrange function $L(x, \mu)$ of problem (Rsvp) the *augmented Lagrange function*

$$\mathcal{L}(x, \mu) = f(x) + \sum_{j=1}^{k} \mu_j \theta_j(g_j(x))$$

(which is the classical Lagrange function of the equivalent problem), we normally get possibility to solve the constrained problem (Rsvp) by the Lagrange Multipliers Scheme (which now becomes augmented).

All this is completely similar to the Augmented Lagrangian Scheme we used in the general case, including possibility to introduce penalty parameter into our augmented scheme for (Rsvp); the "penalized" augmented Lagrangian for (Rsvp) is

$$\mathcal{L}^\rho(x, \mu) = f(x) + \sum_{j=1}^{k} \mu_j \rho^{-1} \theta_j(\rho g_j(x));$$

this corresponds to replacing $\theta_j(\cdot)$ by the rescaled functions $\bar{\theta}_j(s) = \rho^{-1}\theta_j(\rho s)$, which also satisfy the normalization conditions. In the general (nonconvex) case the role of the penalty parameter is to ensure the crucial property (!); besides this, large penalty, as we remember, enforces the Hessian of the dual objective at λ^* to be almost proportional to the unit matrix and thus assures fast local convergence of simple gradient-type methods for solving the dual problem. In our now case, we are not interested in the first of these two issues – due to the convexity of the original problem, (!) is ensured even with small values of ρ; the second advantage of penalization, anyhow, still is important: the larger is ρ, the better is conditioning of the dual objective associated with the augmented Lagrange function $\mathcal{L}^\rho$ and the better are our abilities for fast maximization of this objective.

I should stress that the presented brief outline of the Augmented Lagrangian Multipliers Scheme in Convex Programming is far from being complete; e.g., we did not speak in details on how to solve the dual problem associated with the augmented Lagrangian (now this problem is constrained, although with simple constraints, and simple gradient projection routines are possible, but in no sense best possible options). More detailed considerations of these issues are beyond the scope of this course.

10.4.2 Lagrange Duality and Decomposition

Although the below considerations are not related to *Augmented* Lagrange Multipliers Scheme, I cannot avoid them when speaking on Lagrange Duality and its computational consequences. Thus, let us speak about advantages of the *classical* Lagrange Multipliers Scheme in the case of *separable* problems. In many real-world situations objective and constraints of a Convex Programming program (Rsvp) are *separable*: one can partition the vector of design variables x into non-overlapping sub-vectors $x_1, ..., x_l$ in such a way that

$$f(x) = \sum_{i=1}^{l} f_i(x_i), \;\; g_j(x) = \sum_{i=1}^{l} g_{ji}(x_i).$$

Whenever it is the case, the Lagrange scheme becomes especially attractive due to the following evident property: computation of the dual objective

$$\phi^*(\mu) = \min_x \left[f(x) + \sum_{j=1}^k \mu_j g_j(x) \right]$$

clearly reduces to solving l independent problems

$$(P_i) \quad \min_{x_i} \left[f(x_i) + \sum_{j=1}^k \mu_j g_{ji}(x_i) \right].$$

$i = 1, ..., k$. Note that the computational complexity of an optimization problem normally is superadditive function of the design dimension: it is easier, and usually - much easier, to solve l independent convex problems (P_i) of a given total design dimension n than to solve a single convex program of dimension n. Therefore separability simplifies a lot implementation of the classical Lagrange scheme (note that the main computational effort in this scheme is the one to solve auxiliary primal problems). In any words, Lagrange scheme, in contrast to other Convex Programming techniques, is capable to exploit separability of the objective and the constraints: it allows to *decompose* the original problem into a collection of independent problems which "interact" with each other only via the "coordinating" dual problem (CnvD).

In fact similar decomposition approach can be carried out also in the case when (Rsvp) contains, besides separable objective and constraints, a number of constraints depending each of its own part x_i of the variables. Such a problem can be written down as

$$\text{(CnvPS)} \quad \min_x \left\{ \sum_{i=1}^l f_i(x_i) : \sum_{i=1}^l g_{ji}(x) \le 0,\ j = 1, ..., k,\ x_i \in X_i \right\},$$

where X_i are the convex sets given by those inequalities which depend on x_i solely.

The corresponding version of Lagrange Duality says that the natural dual problem is

$$\text{(CnvDS)} \quad \sum_{i=1}^l \phi_i^*(\mu) \to \max,\ \ \mu \ge 0,$$

where

$$\phi_i^*(\mu) = \min_{x_i \in X_i} \left[f_i(x_i) + \sum_{j=1}^k \mu_j g_{ji}(x_i) \right];$$

the difference with the standard duality is that now minimization over x_i in the formulae for dual objective is taken over the set X_i given by the constraints depending on x_i solely, not over the entire space.

The meaning of the words "(CnvDS) is natural dual to (CnvPS)" is given by the corresponding version of the Duality Theorem:

(*) *If (CnvPS) is solvable and satisfies the Slater condition, then (CnvDS) (which always is a Convex Programming problem) also is solvable, the optimal values in the problems are equal to each other, and the pair (x^*, μ^*) of feasible primal and dual solutions to the problems is comprised of optimal solutions if and only if (x^*, λ^*) is saddle point of the classical Lagrange function $L(x, \mu)$ of (CnvPS) on the set $X \equiv X_1 \times ... \times X_l$ of values of x and the set $\mathbf{R}^k_+ = \{\mu \in \mathbf{R}^k \mid \mu \ge 0\}$ of values of μ, i.e., if $L(x, \mu^*)$ attains its minimum over $x \in X$ at $x = x^*$, while the function $L(x^*, \mu)$ attains its maximum over $\mu \ge 0$ at $\mu = \mu^*$.*

As a corollary, we get that if μ^* is optimal solution to (CnvDS), then the components x_i^* of (any) optimal solution to (CnvPS) are among the minimizers of the Lagrange functions

$$L_i(x_i, \mu^*) = f_i(x) + \sum_{j=1}^k \mu_j^* g_{ji}(x_i)$$

over $x_i \in X_i$. Under assumptions of strong convexity similar to those indicated in the previous subsection, this information is sufficient to restore, given μ^*, the components of optimal primal solution x^*. In the general case we again might meet with the difficulty that there are many minimizers of $L_i(x_i, \mu^*)$ over $x_i \in X_i$, and only some of these minimizers can be combined into optimal solution to (CnvPS). How to overcome this difficulty, this is another story which goes beyond the bounds of this lecture; what should be stressed, anyhow, is that the outlined decomposition scheme allows to obtain important, and sometimes complete information on the optimal solution to (CnvPS).

The Decomposition approach is especially attractive when l and the total number of constraints participating in the description of the sets X_i are large, and there is relatively small number of separable "linking constraints" $g_j(x) \leq 0$. Indeed, in this case the dual problem (CnvDS) is of small dimension, which simplifies it computationally. At the same time, to compute the objective of (CnvDS) at a given μ, one has to solve l independent problems of minimizing $L_i(x_i, \mu)$ over $x_i \in X_i$. This latter task can be achieved by parallel computations, and even with serial computations it is, as I have already mentioned, much easier to solve many "small" independent problems than a single "large" one.

The outlined Decomposition scheme has very nice economical interpretation. Imagine that certain company is comprised of basically independent units; the only interconnections between the units are that all the units consume a number of resources like money, energy, etc., which are controlled by the company. Let us interpret x_i as the action of i-th unit, X_i as the set of actions allowed for the unit by its own technological constraints. Let, further, $g_{ji}(x_i)$ be the amount of resource j which is required by unit i to implement action x_i, and let $-f_i(x_i)$ be the profit the unit will get carrying out this action. The company is interested to maximize the total profit of the units

$$-f(x) \equiv \sum_{i=1}^{l} (-f_i(x_i))$$

under the constraints

$$\sum_{i=1}^{l} g_{ji}(x_i) \leq b_j,\ j = 1, ..., m,$$

on the total amount of the resources consumed by the units, and, of course, the constraints

$$x_i \in X_i,\ i = 1, ..., l.$$

This is certain optimization problem, and a straightforward way to solve it would be: to "lift" to the level of the company all the data (sets X_i, functions g_{ji}, f_i), to solve at this level the resulting problem and to enforce somehow the units to carry out the resulting directives x_i^*. This completely centralized approach is very unreliable, due to the necessity in sending back and forth huge amounts of information, and is very costly computationally. A good alternative could be *decentralized* approach, where the company declares some "internal prices" on the common resources and says to the units:
"I can sell to you as many resources as you wish at such and such prices. Please order the resources and say what will be your profit".

The units will try to maximize, given prices of the resources, their profit

$$-f_i(x_i) - \sum_{j=1}^{k} \mu_j g_{ji}(x_i)$$

over $x_i \in X_i$, i.e., minimize their losses $L_i(x_i, \mu)$; their orders will allow the company to compute the objective of the dual problem (CnvDS), which for the case of nonzero right hand sides in the resource inequality constraints clearly is

$$\phi^*(\mu) = \sum_{j=1}^{k} L_i(x_i^*(\mu), \mu) - \sum_{j=1}^{m} \mu_j b_j.$$

In fact the vectors of the resources ordered by the units allow the company to compute the gradient of the dual objective at the current μ (or a supergradient, if the dual objective is nonsmooth). With this

information, the company can update the prices on the resources, applying certain method for concave minimization of the dual objective on the nonnegative orthant, ask the units to update accordingly their decisions, and to repeat this procedure until the dual objective will be maximized, thus solving – in parallel and distributed fashion – the problem of maximizing total profit under given constraints on the common resources. This is exactly what is Decomposition does.

It is time now to explain why in the Decomposition scheme we deal with the classical Lagrange function, not an augmented one. The reason is that the "augmenting" $g_j \mapsto \theta_j(g_j)$ of a separable constraint $g_j(x) \leq 0$ destroys separability. The price for using the classical Lagrangian is typical nonsmoothness of the dual objective, and in fact Decomposition is the main source of *nonsmooth* convex optimization programs.

Lecture 11

Sequential Quadratic Programming

This last lecture of the course is devoted to the *Sequential Quadratic Programming* methods – the methods which are now thought to be the most efficient for practical solution of general-type (i.e., nonconvex) constrained optimization problems.

The idea underlying the methods is to solve directly the KKT system of the problem under consideration by a Newton-type iterative process. In order to get the idea, let us start with the equality constrained case.

11.1 SQP methods: Equality Constrained case

Consider an equality constrained optimization problem

$$\text{(ECP)} \quad \min_x \left\{ f(x) : h(x) = \begin{pmatrix} h_1(x) \\ ... \\ h_m(x) \end{pmatrix} = 0 \right\} \quad [x \in \mathbf{R}^n]$$

and let us write down the corresponding Karush-Kuhn-Tucker system:

$$\begin{array}{ll} \text{(KKT)} & \\ \nabla_x L(x,\lambda) & \equiv \quad \nabla f(x) + [\nabla h(x)]^T \lambda = 0 \ , \\ \nabla_\lambda L(x,\lambda) & \equiv \quad h(x) = 0, \end{array}$$

where

$$L(x,\lambda) = f(x) + \lambda^T h(x) \equiv f(x) + \sum_{i=1}^m \lambda_i h_i(x)$$

is the Lagrangian of (ECP). As we remember from Lecture 8, any locally optimal solution x^* of (ECP) which is a regular point of the system of constraints (i.e., is such that the gradients of constraints taken at this point are linearly independent) can be extended by properly chosen $\lambda = \lambda^*$ to a solution of (KKT).

Now, (KKT) is a system of $n+m$ equations with $n+m$ unknowns x, λ, and we can try to solve this system with the Newton method. To the moment we know what is the Newton *minimization* method, not the Newton method for solving systems of equations; let us look what is this latter routine.

11.1.1 Newton method for systems of equations

11.1.1.1 The method

Given a system of N nonlinear scalar equations with N unknowns

$$\text{(E)} \qquad P(u) \equiv \begin{pmatrix} p_1(u) \\ ... \\ p_N(u) \end{pmatrix} = 0,$$

with continuously differentiable real-valued functions p_i, one can try to solve this system as follows. Given current approximate solution $\bar{u}$, let us linearize the system at this point, i.e., replace the actual nonlinear system by the linear one

$$[P(u) \approx] \quad P(\bar{u}) + P'(\bar{u})(u - \bar{u}) \equiv \begin{pmatrix} p_1(\bar{u}) + [\nabla p_1(\bar{u})]^T(u - \bar{u}) \\ ... \\ p_N(\bar{u}) + [\nabla p_N(\bar{u})]^T(u - \bar{u}) \end{pmatrix} = 0.$$

Assuming the $N \times N$ matrix $P'(\bar{u})$ nonsingular, we can write down the solution to the linearized system:

$$\bar{u}^+ = \bar{u} - [P'(\bar{u})]^{-1} P(\bar{u}).$$

The vector

$$\bar{u}^+ - \bar{u} \equiv -[P'(\bar{u})]^{-1} P(\bar{u})$$

is called the *Newton displacement*.

Now we can look at $\bar{u}^+$ as at our new approximate solution and iterate the indicated updating, thus coming to the *Newton method for solving (E)* – to the recurrence

$$u_t = u_{t-1} - [P'(u_{t-1})]^{-1} P(u_{t-1}). \tag{11.1.1}$$

Note that the basic Newton method for unconstrained minimization of a smooth function f (Lecture 4) is nothing but the above recurrence applied to the Fermat equation[1)]

$$\text{(F)} \qquad P(x) \equiv \nabla f(x) = 0.$$

11.1.1.2 Local quadratic convergence

Same as in the particular case of equation (F), in the case of general equation (E) the Newton method possesses fast local convergence:

Proposition 11.1.1 [Local superlinear/quadratic convergence of the Newton method for solving systems of equations]
Let u^* be a solution to (E), and assume that this solution is nondegenerate, i.e., that the matrix $P'(u^*)$ is nonsingular. Then the Newton method (11.1.1) locally superlinearly converges to u^*: there exists a neighbourhood U of u^* such that the recurrence (11.1.1), being started at a point $u_0 \in U$, is well defined (i.e., the required inverse matrices do exist), keeps the iterates in U and ensures that

$$|u_t - u^*| \le C_\kappa \kappa^t, \ t = 0, 1, ...$$

for every $\kappa > 0$.

If, in addition, P is twice continuously differentiable in a neighbourhood of u^*, then the indicated convergence is quadratic:

$$|u_t - u^*| \le C|u_{t-1} - u^*|^2$$

for all t and some $C < \infty$.

The proof of this proposition repeats word by word the one of the similar statement from Lecture 4, and you are welcome to prove the statement by yourself.

Now let us look how all this works when (E) is the Karush-Kuhn-Tucker system of the equality constrained problem (ECP).

[1)] to save words, we shall use the word "equation" also for systems of scalar equations, interpreting such a system as a single vector equation. It always will be clear which equation – a vector or a scalar one – is meant.

11.1.2 Solving (KKT) by the Newton method

When applying the Newton method to (KKT), we should answer the following crucial question:

- When a KKT point (x^*, λ^*) of problem (ECP) is a nondegenerate solution to the system (KKT)?

  [when it is so, we may hope to get at least *locally* converging routine; on the other hand, when the corresponding matrix P' is singular at (x^*, λ^*), we have small hope on convergence, not speaking of the unpleasant possibility to meet with singular P' also at a point close to (x^*, λ^*): if this latter point will be our iterate, how could we define the Newton displacement?]

Our local goal is to answer the above question and to present important interpretation of the Newton displacement in terms of certain auxiliary quadratic optimization problem with linear equality constraints. This interpretation will allow us to extend the approach from the case of equality constrained problem (ECP) to the general constrained case.

To get convenient notation, we set

$$P(x,\lambda) = \nabla_{x,\lambda} L(x,\lambda) = \begin{pmatrix} \nabla_x L(x,\lambda) \equiv \nabla f(x) + [\nabla h(x)]^T \lambda \\ \nabla_\lambda L(x,\lambda) \equiv h(x) \end{pmatrix}; \tag{11.1.2}$$

in this notation, system (KKT) is exactly (E). Note that

$$P'(x,\lambda) = \begin{pmatrix} \nabla_x^2 L(x,\lambda) & [\nabla h(x)]^T \\ \nabla h(x) & 0 \end{pmatrix} \tag{11.1.3}$$

11.1.2.1 Nonsingularity of KKT points

We start with the following important observation:

Proposition 11.1.2 *Let x^* be a nondegenerate solution to (ECP)* (Definition 4.2.1)*, and let λ^* be the corresponding vector of Lagrange multipliers. Then the matrix $P'(x^*, \lambda^*)$ is nonsingular.*

Proof. Denoting, as usual, $H = \nabla_x^2 L(x^*, \lambda^*)$, $A = \nabla h(x^*)$, we have

$$P' = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix};$$

we should prove that $P'h = 0$, h being $(n+m)$-dimensional vector, implies $h = 0$. Partitioning h into n-dimensional fragment v and m-dimensional fragment w, we get

$$0 = P' \begin{pmatrix} v \\ w \end{pmatrix} \equiv \begin{pmatrix} Hv + A^T w \\ Av \end{pmatrix},$$

whence

$$Hv + A^T w = 0; \quad Av = 0.$$

The second equation says that v is in the kernel of $A = \nabla h(x^*)$, i.e., is in the plane T tangent to the feasible surface at the point x^*. Multiplying the first equation from the left by v^T and taking into account that $v^T A^T w = (Av)^T w = 0$ by the second equation, we get $v^T H v = 0$. Since x^* is a nondegenerate local solution to (ECP), $H = \nabla_x^2 L(x^*, \lambda^*)$ is positive definite on T; consequently, $v \in T$ and $v^T H v = 0$ imply that $v = 0$.

Since $v = 0$, the first of our two equations implies that $A^T w = 0$; but x^* is regular for the constraints of (ECP), i.e., the columns of the matrix $A^T = [\nabla h(x^*)]^T$ – these columns are exactly the gradients of the constraints at the point x^* – are linearly independent; since the columns of A^T are linearly independent, their combination $A^T w$ can be zero only if the vector w of coefficients in the combination is zero. Thus, both v and w are zero. ∎

11.1.2.2 Structure and interpretation of the Newton displacement

Let us look what is the Newton system

$$P'(u)d = -P(u)$$

– the system defining the Newton displacement – for the particular case when the system (E) of nonlinear equations is the Karush-Kuhn-Tucker system (KKT) of (ECP). In the case in question P is given by (11.1.2), while P' is given by (11.1.3). Partitioning the vector of unknowns d of the Newton system into x-part dx and λ-part $d\lambda$, we can rewrite the Newton system as

$$\begin{array}{rcl} [\nabla_x^2 L(\bar{x},\bar{\lambda})]dx + [\nabla h(\bar{x})]^T d\lambda & = & -\nabla f(\bar{x}) - [\nabla h(\bar{x})]^T\lambda \\ [\nabla h(\bar{x})]d\lambda & = & -h(\bar{x}) \end{array},$$

$(\bar{x},\bar{\lambda})$ being the current iterate to be updated by the Newton step.

It is convenient to pass from the unknowns dx and $d\lambda$ to dx and $\lambda^+ = \bar{\lambda} + d\lambda$ (note that λ^+ is the λ-component of the Newton iterate of $(\bar{x},\bar{\lambda})$). The new unknowns are given by the system

$$\begin{array}{rcl} [\nabla_x^2 L(\bar{x},\bar{\lambda})]dx + [\nabla h(\bar{x})]^T\lambda^+ & = & -\nabla f(\bar{x}) \\ [\nabla h(\bar{x})]dx & = & -h(\bar{x}) \end{array} \qquad (11.1.4)$$

The resulting form of the Newton system is very instructive. To see this, let us for the time being forget about the Newton method for solving the KKT system of problem (ECP) and look at the situation from a little bit different viewpoint. Let x^* be a nondegenerate solution to (ECP) and λ^* be the corresponding vector of Lagrange multipliers. Then

$$\nabla_x L(x^*,\lambda^*) = 0$$

and the Hessian $\nabla_x^2 L(x^*,\lambda^*)$ of the Lagrange function is positive definite *on the plane T tangent to the feasible surface of (ECP) at the point x^**. In other words, the Lagrange function regarded as function of x (with λ being set to λ^*) attains at $x = x^*$ its nondegenerate local minimum *on the plane T*. If we knew the tangent plane T (and λ^*) in advance, we could find x^* by applying to $L(x,\lambda^*)$ any method of unconstrained minimization, not in the entire space, of course, but in the plane T. Let us look what would happen if we would use in this scheme the Newton minimization method. How we should act? We should replace our current objective – the function $L(\cdot,\lambda^*)$ – by its quadratic expansion at the current iterate $\bar{x}$, i.e., by the function

$$L(\bar{x},\lambda^*) + (dx)^T\nabla_x L(\bar{x},\lambda^*) + \frac{1}{2}(dx)^T\nabla_x^2 L(\bar{x},\lambda^*)dx \qquad (11.1.5)$$

and to minimize this quadratic approximation over displacements dx such that $\bar{x} + dx \in T$; the new iterate would be then $\bar{x} + dx$, where dx is given by the aforementioned minimization.

Now, actually we do not know T and λ^*; all we have is the current approximation $(\bar{x},\bar{\lambda})$ of (x^*,λ^*). With this approximation, we can imitate the latter construction as follows:

- use, instead of λ^*, its current approximate $\bar{\lambda}$;
- use, instead of T, the plane $\bar{T}$ given by *linearization* of the constraints $h(x) = 0$ at the current iterate $\bar{x}$:

$$\bar{T} = \{y = \bar{x} + dx \mid [\nabla h(\bar{x})]dx + h(\bar{x}) = 0\}.$$

Thus, the above idea – to improve current iterate $\bar{x}$ by applying a single step of the Newton minimization of the Lagrange function over the plane T – could be implemented as follows: we define dx as the solution to the quadratic minimization problem with linear equality constraints:

$$\begin{array}{ll} (*) & \\ & L(\bar{x},\bar{\lambda}) + (dx)^T\nabla_x L(\bar{x},\bar{\lambda}) + \frac{1}{2}(dx)^T\nabla_x^2 L(\bar{x},\bar{\lambda})dx \quad \to \quad \min \\ \text{s.t.} & \\ & [\nabla h(\bar{x})]dx \quad = \quad -h(\bar{x}) \end{array}$$

and define the new x-iterate as

$$x^+ = \bar{x} + dx.$$

Note that the problem can be a little bit simplified: recalling the definition of the Lagrange function, we see that the linear part

$$L(\bar{x}, \bar{\lambda}) + (dx)^T \nabla_x L(\bar{x}, \bar{\lambda})$$

of the objective in the problem is equal to

$$f(\bar{x}) + (dx)^T \nabla f(\bar{x}) + \left[\bar{\lambda}^T h(\bar{x}) + \bar{\lambda}^T [\nabla h(\bar{x})] dx\right];$$

the quantity in the brackets is constant on the feasible plane of the problem and therefore can be eliminated from the objective without varying the solution. Thus, problem (*) is equivalent to

$$\begin{array}{ll} (\mathrm{QP}_{(\bar{x},\bar{\lambda})}) & \\ & f(\bar{x}) + (dx)^T \nabla f(\bar{x}) + +\frac{1}{2}(dx)^T \nabla_x^2 L(\bar{x}, \bar{\lambda}) dx \quad \to \quad \min \\ \text{s.t.} & \\ & [\nabla h(\bar{x})] dx \quad = \quad -h(\bar{x}). \end{array}$$

Problem $(\mathrm{QP}_{(\bar{x},\bar{\lambda})})$ is quite respectable optimization problem, and its solution, if exists, is achieved at a KKT point. Moreover, the solution to the problem does exist, if $(\bar{x}, \bar{\lambda})$ is close enough to (x^*, λ^*). Indeed, if $\bar{x} = x^*, \bar{\lambda} = \lambda^*$, then the objective of the problem is a quadratic form which is strongly convex at the feasible plane T of the problem (recall that x^* is a nondegenerate solution to (ECP)). By continuity reasons, for $(\bar{x}, \bar{\lambda})$ close enough to (x^*, λ^*) the objective in $(\mathrm{QP}_{(\bar{x},\bar{\lambda})})$ also is strongly convex quadratic form on the feasible plane $\bar{T}$ of the problem and therefore attains its minimum on the plane.

Now let us make the following crucial observation (for the sake of brevity, we write (QP) instead of $(\mathrm{QP}_{(\bar{x},\bar{\lambda})})$):

> *Let the Newton system (11.1.4) be a system with nonsingular matrix (we know that it indeed is the case when $(\bar{x}, \bar{\lambda})$ is close enough to (x^*, λ^*)). Then the Newton displacement dx given by the Newton system is the unique KKT point of the problem (QP), and the vector λ^+ given by the Newton system is the vector of Lagrange multipliers associated with the KKT point dx in the problem (QP).*

Indeed, let z be a KKT point of (QP) and μ be the corresponding vector of Lagrange multipliers. The KKT equations for (QP) are (check it!)

$$\begin{array}{rcl} \nabla f(\bar{x}) + \nabla_x^2 L(\bar{x}, \bar{\lambda}) z + [\nabla h(\bar{x})]^T \mu & = & 0, \\ [\nabla h(\bar{x})] z & = & -h(\bar{x}); \end{array}$$

but this is exactly the Newton system (11.1.4).

Now we understand what the Newton method from the previous section does in the optimization terms. The optimal solution is the minimizer of the Lagrange function on the tangent plane T to the feasible surface of the problem, the Lagrange multipliers being set to their optimal values. The method, at each step, replaces this the tangent plane with the plane $\bar{T}$ given by linearizations of the constraints at the current iterate, replaces "exact" multipliers in the Lagrange function with their current estimate and performs a single Newton step of minimization of the resulting function over $x \in \bar{T}$, thus generating the new x-iterate. As a byproduct of this Newton minimization step (this step solves quadratic minimization problem (QP) with linear equality constraints) we get the vector of Lagrange multipliers for (QP), and this vector is our new estimate of the vector λ^* of Lagrange multipliers of the original problem.

The presented interpretation of the method explains why it is called "Sequential Quadratic Programming"; the name comes from the fact that at each step of the method we in fact solve certain Quadratic Programming problem with linear equality constraints. There is another name of the method – the *Projected Lagrangian Scheme*, and this second name reflects the origin of our Quadratic Programming problems.

What should be stressed is that the linear term in the objective of (QP) comes from our original objective f [2]; in contrast to this, the *quadratic* term in the objective of (QP) comes from the Lagrange function, not from f. This is quite reasonable: this is the Lagrange function which attains its nondegenerate local minimum on T at x^*, not the actual objective f. For f itself, x^* is nothing but a critical point on T, and it may happen to be the point of maximum, as it is seen from the following example:

$$f(x_1,x_2)=x_2-0.1x_1^2,\ \ h(x)=x_2-x_1^2,\ \ x^*=(0,0).$$

In this example, x^* indeed is a nondegenerate local solution with $\lambda^*=-1$ and $T=\{x\in\mathbf{R}^2\mid x_2=0\}$. The Lagrange function with "correct" value of the Lagrange multiplier is $L(x,\lambda^*)=0.9x_1^2$, and x^* is its nondegenerate local (and even global) minimizer on T. In contrast to this, x^* is nondegenerate global maximizer of the objective f on T.

11.2 The case of general constrained problems

11.2.1 Basic SQP scheme

Given the above interpretation of the Newton method for solving the KKT system of an equality constrained problem, we now can easily understand how the method should be extended onto the case of a general constrained problem

$$\text{(GCP)}\qquad \min_x\left\{f(x):h(x)=\begin{pmatrix}h_1(x)\\ ...\\ h_m(x)\end{pmatrix}=0,\ g(x)=\begin{pmatrix}g_i(x)\\ ...\\ g_k(x)\end{pmatrix}\le 0\right\}.$$

Our starting observation is that if x^* is a nondegenerate solution to (GCP) and (λ^*,μ^*) is the corresponding vector of Lagrange multipliers (λ's are the multipliers for the equality constraints, and μ's - for the inequality ones), then, as it is immediately given by straightforward verification of Definition 4.2.1, x^* is a nondegenerate solution, and λ^*,μ^* are Lagrange multipliers, in the following linearly constrained quadratic programming program:

$$\begin{array}{ll}(\mathrm{QP}^*)&\\ & f(x^*)+(x-x^*)^T\nabla f(x^*)+\frac12(x-x^*)^T\nabla_x^2L(x^*;\lambda^*,\mu^*)(x-x^*)\ \ \to\ \ \min\\ \text{s.t.}&\\ & \begin{array}{rcl}h_i(x^*)+(x-x^*)^T\nabla h_i(x^*) &=& 0,\ i=1,...,m,\\ g_j(x^*)+(x-x^*)^T\nabla g_j(x^*) &\le& 0,\ j=1,...,k,\end{array}\end{array}$$

where

$$L(x;\lambda,\mu)=f(x)+\sum_{i=1}^m\lambda_ih_i(x)+\sum_{j=1}^k\mu_jg_j(x)$$

is the Lagrange function of (GCP).

This observation motivates the following iterative process: given current approximation (x_t,λ_t,μ_t) to (x^*,λ^*,μ^*), we use this approximation to model (QP^*) by the linearly constrained quadratic program

$$\begin{array}{ll}(\mathrm{QP}_t)&\\ & f(x_t)+(dx)^T\nabla f(x_t)+\frac12(dx)^T[\nabla_x^2L(x_t;\lambda_t,\mu_t)]dx\ \ \to\ \ \min\\ \text{s.t.}&\\ & \begin{array}{rcl}[\nabla h(x_t)]dx &=& -h(x_t),\\ [\nabla g(x_t)]dx &\le& -g(x_t).\end{array}\end{array}$$

[2] note, anyhow, that we may deal with the linear term coming from the Lagrange function as well: recall that (QP) is equivalent to (*). The only advantage of (QP) is that the vector of Lagrange multipliers in this problem is exactly the new estimate λ^+ of λ^*, while the vector of Lagrange multipliers in (*) is $\lambda^+-\bar\lambda$

Solving this problem, we get the corresponding displacement dx and, besides this, the vectors λ^+ and μ^+ of the Lagrange multipliers for (QP_t):

$$\begin{aligned}\nabla f(x_t)+\nabla_x^2 L(x_t;\lambda_t,\mu_t)dx+\sum_{i=1}^m \lambda_i^+\nabla h_i(x_t)+\sum_{j=1}^k \mu_j^+\nabla g_j(x_t) &= 0,\\ \mu_j^+[g_j(x_t)+(dx)^T\nabla g_j(x_t)] &= 0,\ j=1,...,k,\\ \mu_j^+ &\geq 0,\ j=1,...,k;\end{aligned}$$

after dx, λ^+, μ^+ are identified, we take as the new iterate the triple

$$x_{t+1}=x_t+dx,\ \ \lambda_{t+1}=\lambda^+,\ \ \mu_{t+1}=\mu^+$$

and loop.

It can be proved that if x^* is a nondegenerate solution to (GCP) and the outlined recurrence is started close enough to (x^*,λ^*,μ^*), then the recurrence is well-defined and converges quadratically to (x^*,λ^*,μ^*).

11.2.2 Quasi-Newton Approximations

A disadvantage of the SQP scheme as developed so far is that the auxiliary quadratic problem (QP_t) may be "bad" – e.g., the quadratic objective in this problem may be below unbounded at the feasible set of (QP_t). This bad thing for sure does not happen in the *equality* constrained case, provided that $x_t,\lambda_t)$ is close enough to (x^*,λ^*), since then the matrix $\nabla^2 L_x(x_t,\lambda_t)$ is positive definite along the approximate tangent plane $\bar{T}$; this favourable situation, however, is ensured only in a neighbourhood of the solution. And in the case of general constraints problem (QP_t) can be bad independently of whether the current iterates are or are not close to $(x^*\lambda^*,\mu^*)$. The situation is very similar to the one we met in the Newton method for unconstrained minimization, where the Hessian of the objective was not obliged to be positive definite at every step, and it caused a lot of troubles.

To overcome the difficulty, in the unconstrained case we simply replaced the current Hessian with its positive definite correction whenever the Hessian was not "well positive definite". The same remedy can be used now: if the objective in (QP_t) turns out to be not strongly convex on the entire space, i.e., if the matrix $\nabla_x^2 L(x_t;\lambda_t,\mu_t)$ is not positive definite, we replace it with its positive definite correction B_t. With this correction, the auxiliary quadratic programming problem which we should solve at a step t of the SQP method becomes

$$\begin{array}{ll}(\mathrm{QP}'_t)&\\ & f(x_t)+(dx)^T\nabla f(x_t)+\frac{1}{2}(dx)^T B_t dx \ \to\ \min\\ \text{s.t.}&\\ & [\nabla h(x_t)]dx = -h(x_t),\\ & [\nabla g(x_t)]dx \leq -g(x_t);\end{array}$$

the x-component of the new iterate is $x_{t+1}=x_t+dx$, while λ_{t+1} and μ_{t+1} are the vectors of Lagrange multipliers of (QP'_t).

The indicated modification (which, as we shall see in a while, still is converging) possesses the following attractive properties:

- problem (QP'_t) does not depend explicitly on the current approximates λ_t and μ_t to the Lagrange multipliers of (GCP) (all the dependence of λ_t,μ_t, if any, is via the choice of the matrix B_t);
- it ensures *global* convergence of a reasonable linesearch version of the SQP scheme (see Section 11.3);

- (QP'_t) becomes a problem with strongly convex quadratic objective and linear constraints; the problem therefore can be efficiently solved in finitely many steps by the active set method (Section 9.1.2).

  It should be also mentioned that if, as it is the case under reasonable regularity assumptions, the Lagrange multipliers μ_{t+1} for inequality constraints in the problem (QP'_t) converge to the actual Lagrange multipliers μ^* for the inequality constraints in (GCP), then for large t the vector μ_t is close to μ_{t+1}. It means that μ_t *allows to predict which inequalities in (QP'_t) are active at the solution to this problem*: for large t, these active inequalities are exactly those which correspond to positive entries of μ_t. It follows that when solving (QP'_t) by the active set method, it makes sense to use this prediction as the initial guess for the active set of (QP'_t). With this policy, eventually (for large enough values of t) (QP'_t) will be solved in *one* step of the active set method (i.e., at the cost of solving a single system of linear equations)! And even for initial values of t, when it may take several steps of the active set method to solve (QP'_t), this prediction used as the initial guess for the actual active set of (QP'_t) typically saves a lot of computations.

- using matrix B_t instead of the Hessian of the Lagrangian, we may apply the Quasi-Newton tactics to approximate this Hessian (more exactly, its positive definite correction) *without explicit usage of the second-order information on the data*

11.3 Linesearch, Merit functions, global convergence

The Sequential Quadratic Programming scheme studied so far possesses nice *local* convergence properties, at least for the case when $B_t = \nabla_x^2 L(x_t; \lambda_t, \mu_t)$: if x^* is a nondegenerate local solution to (GCP) with Lagrange multipliers λ^*, μ^*, and if SQP is initialized by a triple (x_1, λ_1, μ_1) close enough to (x^*, λ^*, μ^*), then the routine quadratically converges to the solution of (GCP). The difficulty, anyhow, is that we have no idea whether the routine possesses *global* convergence, or, better to say, we know for sure that it may fail to converge globally. Indeed, in the simplest case where there is no constraints at all, our routine becomes the Basic Newton method for unconstrained minimization from Lecture 4, and we know from this Lecture that the Newton method, even applied to a fine convex objective, may diverge, if the starting point is too far from the optimal solution.

What we did in the unconstrained case in order to overcome this drawback? The remedy was *linesearch*. More specifically, what we did in our now situation would look as follows:

> given current iterate $u_t = (x_t, \lambda_t, \mu_t)$ and computing dx, λ^+, μ^+ as a solution to (QP'_t) and the Lagrange multipliers associated with this solution, we treat the triple
>
> $$d_{t+1} = (dx, d\lambda = \lambda^+ - \lambda_t, d\mu = \mu^+ - \mu_t) \tag{11.3.1}$$
>
> not as the actual step we should perform from u_t to get the new iterate u_{t+1}, but as a *search direction* only; the new iterate u_{t+1} is obtained from u_t by certain – not necessarily unit – stepsize γ_{t+1} from u_t in the direction d_t:
>
> $$u_{t+1} = u_t + \gamma_{t+1} d_{t+1}, \tag{11.3.2}$$
>
> and the role of linesearch is to choose a reasonable stepsize γ_{t+1}.

In the case of unconstrained optimization we measured the quality of a candidate stepsize by the progress in the objective, so that the linesearch was aimed to minimize the objective along the search ray $\{u_t + \gamma d_{t+1} \mid \gamma \geq 0\}$. In our now situation, it makes no sense to look at the objective alone: what we are interested in, is "movement towards the KKT set of (GCP)", and the description of this set involves both the objective and the constraints.

Thus, what we need is certain "auxiliary objective", certain integrated measure of violation of the relations we would like to ensure (in our case these are the KKT conditions). Choosing somehow this measure, we could use the associated with this measure linesearch in order to choose stepsizes which indeed decrease violations of our our target relations and thus move us towards the solution set.

The general scheme we come to – some policy of choosing sequential search directions plus tactics of choosing stepsizes in these directions, tactics based on minimization of a reasonable auxiliary objective over the search rays – is typical for many iterative methods, and the auxiliary objective in these methods has a special name - it is called the *merit function*. Good choice of a merit function is as important for quality of the overall routine as a good policy of choosing search directions, and in fact both these components of a method should fit each other. The crucial requirement which should be satisfied by these components is as follows:

(!) *if current iterate u is not a solution to the problem in question, then the search direction d associated with u should be descent for the merit function $M(\cdot)$:*

$$M(u+\gamma d) < M(u) \quad \text{for all small enough positive } \gamma.$$

The role of property (!) is clear: if it is violated, then we may arrive at an iterate which is *not* a solution to the problem we want to solve and at the same time is such that *the merit function increases along the search ray associated with the iterate*. In this case the linesearch will return zero stepsize, and the next iterate will be the same as the previous one, so that we shall stay at this "bad" iterate forever. On the other hand, if (!) is satisfied, then the linesearch will ensure nonzero progress in the value of the merit function at each step where the current iterate is not an exact solution, and the method will become *descent* with respect to the merit function. In this case we have good chances to use the Global Convergence Theorem from Lecture 1 to prove convergence of the method.

Now let us apply the outlined general considerations to the case we are interested in, i.e., to the case when the problem of interest is (GCE) and the policy of choosing search directions is given by the Sequential Quadratic Programming scheme. What we need is a merit function which, along with this policy, fits (!).

11.3.1 l_1 merit function

It turns out that an appropriate (and the most common) merit function in our case is

$$M(x) = f(x) + \theta\left[\sum_{i=1}^{m}|h_i(x)| + \sum_{j=1}^{k} g_j^+(x)\right], \tag{11.3.3}$$

where $\theta > 0$ is parameter of the function and, as always,

$$g_j^+(x) = \max[g_j(x), 0]$$

is the positive part of the constraint g_j.

The following lemma justifies the indicated choice of the merit function:

Lemma 11.3.1 *Let x_t be current iterate, let B_t be a positive definite matrix used in (QP'_t), let dx be the solution to the latter problem and $\lambda \equiv \lambda_{t+1}$, $\mu \equiv \mu_{t+1}$ be the Lagrange multipliers associated with (QP'_t). Assume that θ is large enough:*

$$\theta \geq \max\{|\lambda_1|, ..., |\lambda_m|, \mu_1, \mu_2, ..., \mu_k\} \tag{11.3.4}$$

Then either $dx = 0$, and then x_t is a KKT point of (GCP), or $dx \neq 0$, and then dx is a descent direction of the merit function M.

Proof. By the origin of dx, λ and μ we have (values and derivatives of f, h, g are taken at x_t, $B = B_t$):

$$\nabla f + Bdx + [\nabla h]^T\lambda + [\nabla g]^T\mu = 0, \tag{11.3.5}$$

$$h + [\nabla h]dx = 0, \tag{11.3.6}$$

$$g + [\nabla g]dx \leq 0, \tag{11.3.7}$$

and, finally,

$$\mu_j \geq 0, \;\; \mu_j[g_j + (dx)^T \nabla g_j] = 0, \; j = 1, ..., k. \tag{11.3.8}$$

Looking at these relations, we see that if $dx = 0$, then x_t clearly is a KKT point of (GCP). Now let $dx \neq 0$. We have

$$(dx)^T \nabla f =$$

[see (11.3.5)]

$$= -(dx)^T B dx - \sum_{i=1}^{m} \lambda_i (dx)^T \nabla h_i - \sum_{j=1}^{l} \mu_j (dx)^T \nabla g_j =$$

[see (11.3.7) and (11.3.1)]

$$= -(dx)^T B dx + \sum_{i=1}^{m} \lambda_i h_i + \sum_{j=1}^{k} \mu_j g_j \leq$$

[note that $\mu_j \geq 0$]

$$\leq -(dx)^T B dx + \sum_{i=1}^{m} |\lambda_i||h_i| + \sum_{j=1}^{k} \mu_j g_j^+ \leq$$

[see (11.3.4)]

$$\leq -(dx)^T B dx + \theta \left[\sum_{i=1}^{m} |h_i| + \sum_{j=1}^{k} g_j^+ \right]. \tag{11.3.9}$$

Now let $\gamma > 0$. We have

$$M(x_t) - M(x_t + \gamma dx) = [f(x_t) - f(x_t + \gamma dx)] +$$

$$+\theta \left[\sum_{i=1}^{m} [|h_i(x_t)| - |h(x_t + \gamma dx)|] + \sum_{j=1}^{k} [g_j^+(x_t) - g_j^+(x_t + \gamma dx)] \right]. \tag{11.3.10}$$

Denoting by $O_i(\gamma)$ appropriate functions of γ which tend to 0 as $\gamma \to +0$, we have

$$f(x_t) - f(x_t + \gamma dx) \geq -(dx)^T \nabla f + \gamma O_0(\gamma). \tag{11.3.11}$$

Further,

$$h_i(x_t + \gamma dx) = h_i + \gamma (dx)^T \nabla h_i + \gamma O_i(\gamma) =$$

[see (11.3.6)]

$$= (1 - \gamma) h_i + \gamma O_i(\gamma),$$

whence

$$|h_i(x_t + \gamma dx)| \leq (1 - \gamma)|h_i| + \gamma |O_i(\gamma)|, \;\; i = 1, ..., m. \tag{11.3.12}$$

Similarly,

$$g_j(x_t + \gamma dx) = g_j + \gamma (dx)^T \nabla g_j + \gamma O_{m+j}(\gamma) \leq$$

[see (11.3.7)]

$$\leq (1 - \gamma) g_j + \gamma O_{m+j}(\gamma),$$

whence

$$g_j^+(x_t + \gamma dx) \leq (1 - \gamma) g_j^+ + \gamma |O_{m+j}(\gamma)|, \;\; j = 1, ..., k. \tag{11.3.13}$$

Substituting (11.3.11) - (11.3.13) into (11.3.10), we get

$$M(x_t) - M(x_t + \gamma dx) \geq \gamma (dx)^T \nabla f + \gamma \theta \left[\sum_{i=1}^{m} |h_i| + \sum_{j=1}^{k} g_j^+ \right] + \gamma \theta O(\gamma) \geq$$

[see (11.3.9)]

$$\geq \gamma \left[(dx)^T B_t dx + \theta O(\gamma)\right].$$

Since $dx \neq 0$ and B is positive definite, we have $(dx)^T B_t dx > 0$, and since $O(\gamma) \to 0$ as $\gamma \to +0$, we have also

$$(dx)^T B_t dx + \theta O(\gamma) > 0$$

for all small enough positive γ; from the above computation, for these γ the quantity $M(x_t) - M(x_t + \gamma dx)$ is positive, as required. ∎

11.3.2 SQP Algorithm with Merit Function

Our considerations motivate the following

Algorithm 11.3.1 [Generic SQP Algorithm with l_1 Merit Function for (GCP)]
Initialization. *Choose $\theta_1 > 0$, an arbitrary starting point x_1 and set $t = 1$.*
Step t. *Given current iterate x_t, act as follows:*

- *Choose somehow positive definite matrix B_t;*
- *Form linearly constrained quadratic problem (QP'_t) given by x_t and B_t;*
- *Solve (QP'_t), thus obtaining the solution dx and the vectors of Lagrange multipliers $\lambda = \lambda_{t+1}$ and $\mu = \mu_{t+1}$. If $dx = 0$, terminate: x_t is a KKT point of (GCP).*
- *Check whether*

  $$\theta_t \geq \bar{\theta}_t \equiv \max\{|\lambda_1|, ..., |\lambda_m|, \mu_1, ..., \mu_k\}.$$

  if this inequality is satisfied, set $\theta_{t+1} = \theta_t$, otherwise set

  $$\theta_{t+1} = \max[\bar{\theta}_t, 2\theta_t];$$

- *Find the new iterate*

  $$x_{t+1} = x_t + \gamma_{t+1} dx$$

  by linesearch aimed to minimize the merit function

  $$M_{t+1}(x) = f(x) + \theta_{t+1}\left[\sum_{i=1}^{m} |h_i(x)| + \sum_{j=1}^{k} g_j^+(x)\right]$$

  on the search ray $\{x_t + \gamma dx \mid \gamma \geq 0\}$. Replace t with $t+1$ and loop.

We can prove the following Main Theorem:

Theorem 11.3.1 [Global Convergence of the SQP Method with l_1 Merit Function]
Let problem (GCP) be solved by Algorithm 11.3.1, and assume that

- *a) there exists a bounded and closed set $\Omega \subset \mathbf{R}^n$ such that for $x \in \Omega$ the solution set $D(x)$ of the system of linear inequality constraints*

  $$S(x): \quad [\nabla h(x)]dx = -h(x), \ \ [\nabla g(x)]dx \leq -g(x)$$

  with unknowns dx is nonempty, and each vector $dx \in D(x)$ is a regular solution of system $S(x)$ (Definition 4.1.1);
- *b) the trajectory $\{x_t\}$ of the algorithm belongs to Ω and is infinite (i.e., the method does not terminate with exact KKT point of (GCE));*

- *c) the matrices B_t used in the method are uniformly bounded and uniformly positive definite:*

$$cI \le B_t \le CI$$

for all t, with some $0 < c \le C < \infty$.

Then all accumulation points of the trajectory of the method are KKT points of (GCP).

Sketch of the Proof.
1) Let (QP(x, B)) denote the auxiliary linearly constrained quadratic problem of the type (QP$'_t$) associated with an iterate $x_t = x$ and a matrix $B_t = B$, and let $\mathcal{B}$ be the set of positive definite matrices such that

$$cI \le B \le CI.$$

From a) and c) one can extract (this is easy, although dull) that for $x \in \Omega$ and $B \in \mathcal{B}$ the problem (QP(x, B)) has a unique solution $dx(x, B)$ with uniquely defined Lagrange multipliers $\lambda(x, B), \mu(x, B)$, and the triple $(dx(x, B), \lambda(x, B), \mu(x, B))$ is continuous in $(x, B) \in \Omega \times \mathcal{B}$.
2) From 1) and b) it follows that the Lagrange multipliers generated in course of running the method are uniformly bounded; from the description of the method it therefore follows that the parameter θ_t of the merit function is updated only finitely many times. Thus, forgetting about certain finite initial fragment of the trajectory, we may assume that this parameter simply is constant: $\theta_t \equiv \theta$.
3) To prove the Theorem, we should prove that if $\bar{x}$ is the limit of certain subsequence $\{x_{t_p}\}_{p=1}^{\infty}$ of our (bounded) trajectory, then $\bar{x}$ is a KKT point of (GCE). From c) it follows that the matrices B_{t_p} belong to the (clearly compact) set $\mathcal{B}$; passing to a subsequence, we therefore may assume that

$$x_{t_p} \to \bar{x}, \;\; B_{t_p} \to \bar{B} \in \mathcal{B} \;\; \text{as} \;\; p \to \infty.$$

If the solution $d^+ = dx(\bar{x}, \bar{B})$ to problem (QP$(\bar{x}, \bar{B})$) is zero, then $\bar{x}$ is a KKT point of (GCE) by Lemma 11.3.1, as required. Thus, all we need is to lead to a contradiction the assumption that $d^+ \neq 0$. Under this assumption from the convergencies $x_{t_p} \to \bar{x}$, $B_{t_p} \to \bar{B}$, $p \to \infty$, and from 1) it follows that $d_p \equiv dx(x_{t_p}, B_{t_p}) \to d^+ \neq 0$. From the indicated convergencies and the relation $d^+ \neq 0$ by the arguments completely similar to those used in the proof of Lemma 11.3.1 it follows that the progresses $\delta_p \equiv M(x_{t_p}) - M(x_{t_p+1})$ in the values of the merit function at the steps t_p, $p = 1, 2, ...$, are bounded away from zero:

$$\delta_p \ge \delta > 0, \; p = 1, 2, ...$$

This is clearly a contradiction: our process at each step decreases the merit function, and since this function clearly is below bounded on Q, the sum of progresses in its values over all steps is finite, and therefore these progresses go to 0 as $t \to \infty$. ∎

11.3.3 Concluding remarks

There are several essential comments to the presented method.

- In the presentation of the Generic SQP algorithm with l_1 merit function we did not specify how the matrices B_t are generated; Theorem 11.3.1 also is not that specific in this respect (all it asks for is uniform boundedness and uniform positive definiteness of the matrices B_t). Our preceding considerations, however, make it clear that in order to get fast algorithm, one should make B_t "close" to the Hessian of the Lagrange function associated with the current iterate x and the current (coming from the previous auxiliary quadratic programming problem) estimates of the Lagrange multipliers. Of course, if this Hessian, being available, is not "well positive definite", one should replace it by its positive definite correction. And if the second order information is not available at all, we can generate B_t via quasi-Newton approximations similar to those from Lecture 7.

- A weak point of the method in its now form is the possibility for auxiliary problems (QP'_t) to be unfeasible. This indeed may happen, even in the case of feasible problem (GCP). More exactly, if x^* is a nondegenerate solution to (GCP), then the quadratic problems associated with the iterates close enough to x^* are feasible, but if the current iterate is too far from x^*, the quadratic problem can be infeasible. What to do, if an unfeasible auxiliary quadratic problem is met?

  A popular way to overcome this difficulty is to pass from the auxiliary problem (QP'_t) to the "trust region SQP problem" which gives us both the search direction and the stepsize and *always* is feasible and solvable. Namely, given current iterate x_t and positive definite approximation B_t to the Hessian of the Lagrange function, we find the new iterate $x_{t+1} = x_t + dx$ by solving the optimization problem as follows:

$$\begin{aligned} & [f(x_t) + (dx)^T \nabla f(x_t) + \frac{1}{2}(dx)^T B_t dx] \\ & +\theta[\sum_{i=1}^{m} |h_i(x_t) + (dx)^T \nabla h_i(x_t)| \\ & + \sum_{j=1}^{k} \max[0, g_j(x_t) + (dx)^T \nabla g_j(x_t)]] \quad \to \quad \min, \\ \text{s.t.} \quad & -\delta_t \le (dx)_i \le \delta_t,\ i = 1, ..., n. \end{aligned}$$

  This problem can be easily rewritten as a linearly constrained problem with convex quadratic objective. Solving this problem, we take direct care of decreasing the l_1 Merit Function, or, better to say, its analogy obtained when replacing the actual – nonlinear – constraints in the merit function by their linearizations. The "trust region" bounds $-\delta_t \le (dx)_i \le \delta_t$ should be tight enough to make the "analogy" sufficiently close to the actual merit function.

  It should also be mentioned that if (GCP) is a (feasible) program with *convex* constraints (i.e., the equality constraints h_i are linear, and the inequality ones g_j are convex), then auxiliary quadratic problems (QP'_t) for sure are feasible. Indeed, if $\bar{x}$ is an arbitrary point of $\mathbf{R}^n$ and x is a feasible solution to (GCP), then, from linearity of the equality constraints,

$$h(\bar{x}) + [\nabla h(\bar{x})](x - \bar{x}) = h(x) = 0,$$

  and from the Gradient inequality[3)]

$$g(\bar{x}) + [\nabla g(\bar{x})](x - \bar{x}) \le g(x) \le 0,$$

  so that $x - \bar{x}$ is a feasible solution to the quadratic program (QP') associated with $\bar{x}$.

- "Maratos effect". The Merit Function technique ensures global convergence of the SQP scheme at the cost of "spoiling", to some extent, local quadratic convergence of the "pure" SQP scheme from Sections 11.1.2 and 11.2.1. For the sake of definiteness, let us speak about the case of equality constrained problem (ECP). Moreover, let the matrix $H = \nabla_x^2 L(x^*, \lambda^*)$ be the unit matrix (so that H is positive definite on the entire space). Assume that we use the unit matrix also as B_t at all steps of the SQP method; then, as we know from Section 11.1.2, the unit stepsizes γ_t would result in asymptotical quadratic convergence of the routine, provided that it converges at all. And it is easy to demonstrate that, vise versa, "essentially non-unit" stepsizes will prevent the superlinear convergence. Now, what about "asymptotical compatibility" of the unit stepsizes with

[3] which says that for a differentiable convex function $\phi(\cdot)$ one has

$$\phi(u) \ge \phi(v) + (u - v)^T \nabla \phi(u)$$

for all u, v

the linesearch based on the l_1 Merit Function? Is it true that the linesearch will eventually result in nearly unit stepsizes, or is it at least true that the unit stepsizes do not contradict the Merit Function philosophy, i.e., that close to the solution they decrease the merit function? The answer on both these question is negative. There are examples of problems (ECP) where, arbitrarily close to a nondegenerate solution to the problem, the unit stepsizes in the SQP scheme *increase* the l_1 Merit Function (and in fact all natural merit functions). This phenomenon (it was discovered by N. Maratos in '78 and is called the *Maratos effect*) demonstrates that we cannot expect the SQP scheme with merit function based linesearch to possess superlinear asymptotic rate of convergence. There are some ways to avoid this drawback, but these details are beyond the scope of the course.

Appendix A

Prerequisites from Linear Algebra and Analysis

Regarded as mathematical entities, the objective and the constraints in a Mathematical Programming problem are functions of several real variables; therefore before entering the Optimization Theory and Methods, we need to recall several basic notions and facts about the spaces $\mathbf{R}^n$ where these functions live, same as about the functions themselves. The reader is supposed to know most of the facts to follow, so he/she should not be surprised by a "cooking book" style which we intend to use in this Lecture.

A.1 Space $\mathbf{R}^n$: algebraic structure

Basically all events and constructions to be considered will take place in the *space* $\mathbf{R}^n$ *of n-dimensional real vectors*. This space can be described as follows.

A.1.1 A point in $\mathbf{R}^n$

A point in $\mathbf{R}^n$ (called also an *n-dimensional vector*) is an ordered collection $x = (x_1, ..., x_n)$ of n reals, called the *coordinates*, or *components*, or *entries* of vector x; the space $\mathbf{R}^n$ itself is the set of all collections of this type.

A.1.2 Linear operations

$\mathbf{R}^n$ is equipped with two *basic operations*:

- *Addition of vectors.* This operation takes on input two vectors $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ and produces from them a new vector

$$x + y = (x_1 + y_1, ..., x_n + y_n)$$

  with entries which are sums of the corresponding entries in x and in y.

- *Multiplication of vectors by reals.* This operation takes on input a real λ and an n-dimensional vector $x = (x_1, ..., x_n)$ and produces from them a new vector

$$\lambda x = (\lambda x_1, ..., \lambda x_n)$$

  with entries which are λ times the entries of x.

The as far as addition and multiplication by reals are concerned, the arithmetic of $\mathbf{R}^n$ inherits most of the common rules of Real Arithmetic, like $x + y = y + x$, $(x + y) + z = x + (y + z)$, $(\lambda + \mu)(x + y) = \lambda x + \mu x + \lambda y + \mu y$, $\lambda(\mu x) = (\lambda\mu)x$, etc.

A.1.3 Linear subspaces

Linear subspaces in $\mathbf{R}^n$ are, by definition, nonempty subsets of $\mathbf{R}^n$ which are closed with respect to addition of vectors and multiplication of vectors by reals:

$$L \subset \mathbf{R}^n \text{ is a linear subspace } \Leftrightarrow \left\{ \begin{array}{c} L \neq \emptyset; \\ x, y \in L \Rightarrow x + y \in L; \\ x \in L, \lambda \in \mathbf{R} \Rightarrow \lambda x \in L. \end{array} \right.$$

A.1.3.A. Examples of linear subspaces:

1. The entire $\mathbf{R}^n$;
2. The *trivial* subspace containing the single zero vector $0 = (0, ..., 0)$ [1)]; (this vector/point is called also the origin)
3. The set $\{x \in \mathbf{R}^n : x_1 = 0\}$ of all vectors x with the first coordinate equal to zero.

   The latter example admits a natural extension:
4. The set of all solutions to a *homogeneous* (i.e., with zero right hand side) system of linear equations

$$\left\{ x \in \mathbf{R}^n : \begin{array}{rcl} a_{11}x_1 + ... + a_{1n}x_n & = & 0 \\ a_{21}x_1 + ... + a_{2n}x_n & = & 0 \\ ... & & \\ a_{m1}x_1 + ... + a_{mn}x_n & = & 0 \end{array} \right\} \tag{A.1.1}$$

   always is a linear subspace in $\mathbf{R}^n$. This example is "generic", that is, *every* linear subspace in $\mathbf{R}^n$ is the solution set of a (finite) system of homogeneous linear equations, see Proposition A.3.6 below.
5. *Linear span of a set of vectors.* Given a nonempty set X of vectors, one can form a linear subspace $\mathrm{Lin}(X)$, called the *linear span* of X; this subspace consists of all vectors x which can be represented as *linear combinations* $\sum_{i=1}^{N} \lambda_i x_i$ of vectors from X (in $\sum_{i=1}^{N} \lambda_i x_i$, N is an arbitrary positive integer, λ_i are reals and x_i belong to X). Note that

   > $\mathrm{Lin}(X)$ *is the smallest linear subspace which contains* X*: if* L *is a linear subspace such that* $L \supset X$*, then* $L \supset L(X)$ (why?).

   The "linear span" example also is generic:

   > *Every linear subspace in* $\mathbf{R}^n$ *is the linear span of an appropriately chosen finite set of vectors from* $\mathbf{R}^n$.

   (see Theorem A.1.2.(i) below).

A.1.3.B. Sums and intersections of linear subspaces. Let $\{L_\alpha\}_{\alpha \in I}$ be a family (finite or infinite) of linear subspaces of $\mathbf{R}^n$. From this family, one can build two sets:

1. *The sum* $\sum_\alpha L_\alpha$ of the subspaces L_α which consists of all vectors which can be represented as finite sums of vectors taken each from its own subspace of the family;
2. *The intersection* $\bigcap_\alpha L_\alpha$ of the subspaces from the family.

[1)] Pay attention to the notation: we use the same symbol 0 to denote the real zero and the n-dimensional vector with all coordinates equal to zero; these two zeros are not the same, and one should understand from the context (it always is very easy) which zero is meant.

Theorem A.1.1 *Let $\{L_\alpha\}_{\alpha\in I}$ be a family of linear subspaces of $\mathbf{R}^n$. Then*

(i) *The sum $\sum\limits_\alpha L_\alpha$ of the subspaces from the family is itself a linear subspace of $\mathbf{R}^n$; it is the smallest of those subspaces of $\mathbf{R}^n$ which contain every subspace from the family;*

(ii) *The intersection $\bigcap\limits_\alpha L_\alpha$ of the subspaces from the family is itself a linear subspace of $\mathbf{R}^n$; it is the largest of those subspaces of $\mathbf{R}^n$ which are contained in every subspace from the family.*

A.1.4 Linear independence, bases, dimensions

A collection $X = \{x^1, ..., x^N\}$ of vectors from $\mathbf{R}^n$ is called *linearly independent*, if no nontrivial (i.e., with at least one nonzero coefficient) linear combination of vectors from X is zero.

> Example of linearly independent set: the collection of n standard basic orths $e_1 = (1, 0, ..., 0)$, $e_2 = (0, 1, 0, ..., 0)$, ..., $e_n = (0, ..., 0, 1)$.
>
> Examples of linearly dependent sets: (1) $X = \{0\}$; (2) $X = \{e_1, e_1\}$; (3) $X = \{e_1, e_2, e_1+e_2\}$.

A collection of vectors $f^1, ..., f^m$ is called a *basis* in $\mathbf{R}^n$, if

1. The collection is linearly independent;
2. Every vector from $\mathbf{R}^n$ is a linear combination of vectors from the collection (i.e., $\mathrm{Lin}\{f^1, ..., f^m\} = \mathbf{R}^n$).

> Example of a basis: The collection of standard basic orths $e_1, ..., e_n$ is a basis in $\mathbf{R}^n$.
>
> Examples of non-bases: (1) The collection $\{e_2, ..., e_n\}$. This collection is linearly independent, but not every vector is a linear combination of the vectors from the collection; (2) The collection $\{e_1, e_1, e_2, ..., e_n\}$. Every vector is a linear combination of vectors form the collection, but the collection is not linearly independent.

Besides the bases of the entire $\mathbf{R}^n$, one can speak about the bases of linear subspaces:

A collection $\{f^1, ..., f^m\}$ of vectors is called a *basis of a linear subspace L*, if

1. The collection is linearly independent,
2. $L = \mathrm{Lin}\{f^1, ..., f^m\}$, i.e., all vectors f^i belong to L, and every vector from L is a linear combination of the vectors $f^1, ..., f^m$.

In order to avoid trivial remarks, it makes sense to agree once for ever that

> *An empty set of vectors is linearly independent, and an empty linear combination of vectors $\sum\limits_{i\in\emptyset} \lambda_i x_i$ equals to zero.*

With this convention, the trivial linear subspace $L = \{0\}$ also has a basis, specifically, an empty set of vectors.

Theorem A.1.2 (i) *Let L be a linear subspace of $\mathbf{R}^n$. Then L admits a (finite) basis, and all bases of L are comprised of the same number of vectors; this number is called the dimension of L and is denoted by $\dim(L)$.*

> We have seen that $\mathbf{R}^n$ admits a basis comprised of n elements (the standard basic orths). From (i) it follows that *every* basis of $\mathbf{R}^n$ contains exactly n vectors, and the dimension of $\mathbf{R}^n$ is n.

(ii) *The larger is a linear subspace of $\mathbf{R}^n$, the larger is its dimension: if $L \subset L'$ are linear subspaces of $\mathbf{R}^n$, then $\dim(L) \le \dim(L')$, and the equality takes place if and only if $L = L'$.*

> We have seen that the dimension of $\mathbf{R}^n$ is n; according to the above convention, the trivial linear subspace $\{0\}$ of $\mathbf{R}^n$ admits an empty basis, so that its dimension is 0. Since $\{0\} \subset L \subset \mathbf{R}^n$ for every linear subspace L of $\mathbf{R}^n$, it follows from (ii) that the dimension of a linear subspace in $\mathbf{R}^n$ is an integer between 0 and n.

(iii) *Let L be a linear subspace in $\mathbf{R}^n$. Then*

(iii.1) *Every linearly independent subset of vectors from L can be extended to a basis of L;*

(iii.2) *From every spanning subset X for L – i.e., a set X such that $\mathrm{Lin}(X) = L$ – one can extract a basis of L.*

It follows from (iii) that

– every linearly independent subset of L contains at most $\dim(L)$ vectors, and if it contains exactly $\dim(L)$ vectors, it is a basis of L;

– every spanning set for L contains at least $\dim(L)$ vectors, and if it contains exactly $\dim(L)$ vectors, it is a basis of L.

(iv) *Let L be a linear subspace in $\mathbf{R}^n$, and $f^1, ..., f^m$ be a basis in L. Then every vector $x \in L$ admits exactly one representation*

$$x = \sum_{i=1}^{m} \lambda_i(x) f^i$$

as a linear combination of vectors from the basis, and the mapping

$$x \mapsto (\lambda_1(x), ..., \lambda_m(x)) : L \to \mathbf{R}^m$$

is a one-to-one mapping of L onto $\mathbf{R}^m$ which is linear, i.e. for every $i = 1, ..., m$ one has

$$\begin{array}{rcl} \lambda_i(x+y) & = & \lambda_i(x) + \lambda_i(y) \quad \forall (x, y \in L); \\ \lambda_i(\nu x) & = & \nu \lambda_i(x) \quad \forall (x \in L, \nu \in \mathbf{R}). \end{array} \tag{A.1.2}$$

The reals $\lambda_i(x)$, $i = 1, ..., m$, are called the coordinates of $x \in L$ in the basis $f^1, ..., f^m$.

E.g., the coordinates of a vector $x \in \mathbf{R}^n$ in the *standard basis* $e_1, ..., e_n$ of $\mathbf{R}^n$ – the one comprised of the standard basic orths – are exactly the entries of x.

(v) [Dimension formula] *Let L_1, L_2 be linear subspaces of $\mathbf{R}^n$. Then*

$$\dim(L_1 \cap L_2) + \dim(L_1 + L_2) = \dim(L_1) + \dim(L_2).$$

A.1.5 Linear mappings and matrices

A function $\mathcal{A}(x)$ (another name – *mapping*) defined on $\mathbf{R}^n$ and taking values in $\mathbf{R}^m$ is called *linear*, if it preserves linear operations:

$$\mathcal{A}(x+y) = \mathcal{A}(x) + \mathcal{A}(y) \quad \forall (x, y \in \mathbf{R}^n); \quad \mathcal{A}(\lambda x) = \lambda \mathcal{A}(x) \quad \forall (x \in \mathbf{R}^n, \lambda \in \mathbf{R}).$$

It is immediately seen that a linear mapping from $\mathbf{R}^n$ to $\mathbf{R}^m$ can be represented as multiplication by an $m \times n$ matrix:

$$\mathcal{A}(x) = Ax,$$

and this matrix is uniquely defined by the mapping: the columns A_j of A are just the images of the standard basic orths e_j under the mapping $\mathcal{A}$:

$$A_j = \mathcal{A}(e_j).$$

Linear mappings from $\mathbf{R}^n$ into $\mathbf{R}^m$ can be added to each other:

$$(\mathcal{A} + \mathcal{B})(x) = \mathcal{A}(x) + \mathcal{B}(x)$$

and multiplied by reals:

$$(\lambda \mathcal{A})(x) = \lambda \mathcal{A}(x),$$

and the results of these operations again are linear mappings from $\mathbf{R}^n$ to $\mathbf{R}^m$. The addition of linear mappings and multiplication of these mappings by reals correspond to the same operations with the

matrices representing the mappings: adding/multiplying by reals mappings, we add, respectively, multiply by reals the corresponding matrices.

Given two linear mappings $\mathcal{A}(x) : \mathbf{R}^n \to \mathbf{R}^m$ and $\mathcal{B}(y) : \mathbf{R}^m \to \mathbf{R}^k$, we can build their superposition

$$\mathcal{C}(x) \equiv \mathcal{B}(\mathcal{A}(x)) : \mathbf{R}^n \to \mathbf{R}^k,$$

which is again a linear mapping, now from $\mathbf{R}^n$ to $\mathbf{R}^k$. In the language of matrices representing the mappings, the superposition corresponds to matrix multiplication: the $k \times n$ matrix C representing the mapping $\mathcal{C}$ is the product of the matrices representing $\mathcal{A}$ and $\mathcal{B}$:

$$\mathcal{A}(x) = Ax,\ \mathcal{B}(y) = By \Rightarrow \mathcal{C}(x) \equiv \mathcal{B}(\mathcal{A}(x)) = B \cdot (Ax) = (BA)x.$$

Important convention. When speaking about adding n-dimensional vectors and multiplying them by reals, it is absolutely unimportant whether we treat the vectors as the column ones, or the row ones, or write down the entries in rectangular tables, or something else. However, when matrix operations (matrix-vector multiplication, transposition, etc.) become involved, it is important whether we treat our vectors as columns, as rows, or as something else. For the sake of definiteness, *from now on we treat all vectors as column ones*, independently of how we refer to them in the text. For example, when saying for the first time what a vector is, we wrote $x = (x_1, ..., x_n)$, which might suggest that we were speaking about row vectors. We stress that it is <u>not</u> the case, and the only reason for using the notation $x = (x_1, ..., x_n)$ instead of the "correct" one $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ is to save space and to avoid ugly formulas like $f(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix})$ when speaking about functions with vector arguments. After we have agreed that *there is no such thing as a row vector in this Lecture course*, we can use (and do use) without any harm whatever notation we want.

Exercise A.1 *1. Mark in the list below those subsets of $\mathbf{R}^n$ which are linear subspaces, find out their dimensions and point out their bases:*

(a) $\mathbf{R}^n$

(b) $\{0\}$

(c) $\emptyset$

(d) $\{x \in \mathbf{R}^n : \sum_{i=1}^n ix_i = 0\}$

(e) $\{x \in \mathbf{R}^n : \sum_{i=1}^n ix_i^2 = 0\}$

(f) $\{x \in \mathbf{R}^n : \sum_{i=1}^n ix_i = 1\}$

(g) $\{x \in \mathbf{R}^n : \sum_{i=1}^n ix_i^2 = 1\}$

2. It is known that L is a subspace of $\mathbf{R}^n$ with exactly one basis. What is L?

3. Consider the space $\mathbf{R}^{m\times n}$ of $m \times n$ matrices with real entries. As far as linear operations – addition of matrices and multiplication of matrices by reals – are concerned, this space can be treated as certain $\mathbf{R}^N$.

(a) Find the dimension of $\mathbf{R}^{m\times n}$ and point out a basis in this space

(b) In the space $\mathbf{R}^{n\times n}$ of square $n\times n$ matrices, there are two interesting subsets: the set $\mathbf{S}^n$ of symmetric matrices $\{A=[A_{ij}] : A_{ij}=A_{ij}\}$ and the set $\mathbf{J}^n$ of skew-symmetric matrices $\{A=[A_{ij}] : A_{ij}=-A_{ji}\}$.

i. Verify that both $\mathbf{S}^n$ and $\mathbf{J}^n$ are linear subspaces of $\mathbf{R}^{n\times n}$

ii. Find the dimension and point out a basis in $\mathbf{S}^n$

iii. Find the dimension and point out a basis in $\mathbf{J}^n$

iv. What is the sum of $\mathbf{S}^n$ and $\mathbf{J}^n$? What is the intersection of $\mathbf{S}^n$ and $\mathbf{J}^n$?

A.2 Space $\mathbf{R}^n$: Euclidean structure

So far, we were interested solely in the algebraic structure of $\mathbf{R}^n$, or, which is the same, in the properties of the *linear* operations (addition of vectors and multiplication of vectors by scalars) the space is endowed with. Now let us consider another structure on $\mathbf{R}^n$ – the *standard Euclidean structure* – which allows to speak about distances, angles, convergence, etc., and thus makes the space $\mathbf{R}^n$ a much richer mathematical entity.

A.2.1 Euclidean structure

The standard Euclidean structure on $\mathbf{R}^n$ is given by the *standard inner product* – an operation which takes on input two vectors x, y and produces from them a real, specifically, the real

$$\langle x,y\rangle \equiv x^T y = \sum_{i=1}^n x_i y_i$$

The basic properties of the inner product are as follows:

1. [bi-linearity]: The real-valued function $\langle x,y\rangle$ of two vector arguments $x,y\in\mathbf{R}^n$ is linear with respect to every one of the arguments, the other argument being fixed:

$$\begin{array}{rcll}\langle \lambda u+\mu v, y\rangle &=& \lambda\langle u,y\rangle+\mu\langle v,y\rangle & \forall(u,v,y\in\mathbf{R}^n,\lambda,\mu\in\mathbf{R})\\ \langle x,\lambda u+\mu v\rangle &=& \lambda\langle x,u\rangle+\mu\langle x,v\rangle & \forall(x,u,v\in\mathbf{R}^n,\lambda,\mu\in\mathbf{R})\end{array}$$

2. [symmetry]: The function $\langle x,y\rangle$ is symmetric:

$$\langle x,y\rangle = \langle y,x\rangle \quad \forall(x,y\in\mathbf{R}^n).$$

3. [positive definiteness]: The quantity $\langle x,x\rangle$ always is nonnegative, and it is zero if and only if x is zero.

Remark A.2.1 The outlined 3 properties – bi-linearity, symmetry and positive definiteness – form a definition of an Euclidean inner product, and there are infinitely many different from each other ways to satisfy these properties; in other words, there are infinitely many different Euclidean inner products on $\mathbf{R}^n$. The standard inner product $\langle x,y\rangle = x^T y$ is just a particular case of this general notion. Although in the sequel we normally work with the standard inner product, the reader should remember that the facts we are about to recall are valid for all Euclidean inner products, and not only for the standard one.

The notion of an inner product underlies a number of purely algebraic constructions, in particular, those of *inner product representation of linear forms* and of *orthogonal complement*.

A.2.2 Inner product representation of linear forms on $\mathbf{R}^n$

A *linear form* on $\mathbf{R}^n$ is a real-valued function $f(x)$ on $\mathbf{R}^n$ which is additive ($f(x+y)=f(x)+f(y)$) and homogeneous ($f(\lambda x)=\lambda f(x)$)

Example of linear form: $f(x)=\sum_{i=1}^{n} ix_i$

Examples of non-linear functions: (1) $f(x)=x_1+1$; (2) $f(x)=x_1^2-x_2^2$; (3) $f(x)=\sin(x_1)$.

When adding/multiplying by reals linear forms, we again get linear forms (scientifically speaking: "linear forms on $\mathbf{R}^n$ form a linear space"). *Euclidean structure allows to identify linear forms on $\mathbf{R}^n$ with vectors from $\mathbf{R}^n$*:

Theorem A.2.1 *Let $\langle\cdot,\cdot\rangle$ be a Euclidean inner product on $\mathbf{R}^n$.*

(i) *Let $f(x)$ be a linear form on $\mathbf{R}^n$. Then there exists a uniquely defined vector $f\in\mathbf{R}^n$ such that the form is just the inner product with f:*

$$f(x)=\langle f,x\rangle \quad \forall x$$

(ii) *Vice versa, every vector $f\in\mathbf{R}^n$ defines, via the formula*

$$f(x)\equiv\langle f,x\rangle,$$

a linear form on $\mathbf{R}^n$;

(iii) *The above one-to-one correspondence between the linear forms and vectors on $\mathbf{R}^n$ is linear: adding linear forms (or multiplying a linear form by a real), we add (respectively, multiply by the real) the vector(s) representing the form(s).*

A.2.3 Orthogonal complement

An Euclidean structure allows to associate with a linear subspace $L\subset\mathbf{R}^n$ another linear subspace $L^\perp$ – the *orthogonal complement* (or the *annulator*) of L; by definition, $L^\perp$ consists of all vectors which are orthogonal to every vector from L:

$$L^\perp=\{f:\langle f,x\rangle=0 \quad \forall x\in L\}.$$

Theorem A.2.2 (i) *Twice taken, orthogonal complement recovers the original subspace: whenever L is a linear subspace of $\mathbf{R}^n$, one has*

$$(L^\perp)^\perp=L;$$

(ii) *The larger is a linear subspace L, the smaller is its orthogonal complement: if $L_1\subset L_2$ are linear subspaces of $\mathbf{R}^n$, then $L_1^\perp\supset L_2^\perp$*

(iii) *The intersection of a subspace and its orthogonal complement is trivial, and the sum of these subspaces is the entire $\mathbf{R}^n$:*

$$L\cap L^\perp=\{0\},\quad L+L^\perp=\mathbf{R}^n.$$

Remark A.2.2 From Theorem A.2.2.(iii) and the Dimension formula (Theorem A.1.2.(v)) it follows, first, that for every subspace L in $\mathbf{R}^n$ one has

$$\dim(L)+\dim(L^\perp)=n.$$

Second, every vector $x\in\mathbf{R}^n$ admits a unique decomposition

$$x=x_L+x_{L^\perp}$$

into a sum of two vectors: the first of them, x_L, belongs to L, and the second, $x_{L^\perp}$, belongs to $L^\perp$. This decomposition is called the *orthogonal decomposition* of x *taken with respect to* $L, L^\perp$; x_L is called the *orthogonal projection* of x onto L, and $x_{L^\perp}$ – the orthogonal projection of x onto the orthogonal complement of L. Both projections depend on x linearly, for example,

$$(x+y)_L=x_L+y_L,\quad (\lambda x)_L=\lambda x_L.$$

The mapping $x\mapsto x_L$ is called the *orthogonal projector* onto L.

A.2.4 Orthonormal bases

A collection of vectors $f^1, ..., f^m$ is called *orthonormal* w.r.t. Euclidean inner product $\langle \cdot, \cdot \rangle$, if distinct vector from the collection are orthogonal to each other:

$$i \neq j \Rightarrow \langle f^i, f^j \rangle = 0$$

and inner product of every vector f^i with itself is unit:

$$\langle f^i, f^i \rangle = 1, \ i = 1, ..., m.$$

Theorem A.2.3 (i) *An orthonormal collection $f^1, ..., f^m$ always is linearly independent and is therefore a basis of its linear span $L = \mathrm{Lin}(f^1, ..., f^m)$* (such a basis in a linear subspace is called *orthonormal*). *The coordinates of a vector $x \in L$ w.r.t. an orthonormal basis $f^1, ..., f^m$ of L are given by explicit formulas:*

$$x = \sum_{i=1}^{m} \lambda_i(x) f^i \Leftrightarrow \lambda_i(x) = \langle x, f^i \rangle.$$

Example of an orthonormal basis in $\mathbf{R}^n$: The standard basis $\{e_1, ..., e_n\}$ is orthonormal *with respect to the standard inner product* $\langle x, y \rangle = x^T y$ on $\mathbf{R}^n$ (but is not orthonormal w.r.t. other Euclidean inner products on $\mathbf{R}^n$).

Proof of (i): Taking inner product of both sides in the equality

$$x = \sum_j \lambda_j(x) f^j$$

with f^i, we get

$$\begin{array}{rcll} \langle x, f_i \rangle & = & \langle \sum\limits_j \lambda_j(x) f^j, f^i \rangle & \\ & = & \sum\limits_j \lambda_j(x) \langle f^j, f^i \rangle & \text{[bilinearity of inner product]} \\ & = & \lambda_i(x) & \text{[orthonormality of } \{f^i\}] \end{array}$$

Similar computation demonstrates that if 0 is represented as a linear combination of f^i with certain coefficients λ_i, then $\lambda_i = \langle 0, f^i \rangle = 0$, i.e., all the coefficients are zero; this means that an orthonormal system is linearly independent.

(ii) *If $f^1, ..., f^m$ is an orthonormal basis in a linear subspace L, then the inner product of two vectors $x, y \in L$ in the coordinates $\lambda_i(\cdot)$ w.r.t. this basis is given by the standard formula*

$$\langle x, y \rangle = \sum_{i=1}^{m} \lambda_i(x) \lambda_i(y).$$

Proof:

$$\begin{array}{rcll} x & = & \sum\limits_i \lambda_i(x) f^i, \ y = \sum\limits_i \lambda_i(y) f^i & \\ \Rightarrow \langle x, y \rangle & = & \langle \sum\limits_i \lambda_i(x) f^i, \sum\limits_i \lambda_i(y) f^i \rangle & \\ & = & \sum\limits_{i,j} \lambda_i(x) \lambda_j(y) \langle f^i, f^j \rangle & \text{[bilinearity of inner product]} \\ & = & \sum\limits_i \lambda_i(x) \lambda_i(y) & \text{[orthonormality of } \{f^i\}] \end{array}$$

(iii) *Every linear subspace L of $\mathbf{R}^n$ admits an orthonormal basis; moreover, every orthonormal system $f^1, ..., f^m$ of vectors from L can be extended to an orthonormal basis in L.*

Important corollary: *All Euclidean spaces of the same dimension are "the same". Specifically, if L is an m-dimensional space in a space $\mathbf{R}^n$ equipped with an Euclidean inner product $\langle\cdot,\cdot\rangle$, then there exists a one-to-one mapping $x \mapsto A(x)$ of L onto $\mathbf{R}^m$ such that*

- *The mapping preserves linear operations:*

$$A(x+y) = A(x) + A(y) \quad \forall (x, y \in L); A(\lambda x) = \lambda A(x) \quad \forall (x \in L, \lambda \in \mathbf{R});$$

- *The mapping converts the $\langle\cdot,\cdot\rangle$ inner product on L into the standard inner product on $\mathbf{R}^m$:*

$$\langle x, y\rangle = (A(x))^T A(y) \quad \forall x, y \in L.$$

Indeed, by (iii) L admits an orthonormal basis $f^1, ..., f^m$; using (ii), one can immediately check that the mapping

$$x \mapsto A(x) = (\lambda_1(x), ..., \lambda_m(x))$$

which maps $x \in L$ into the m-dimensional vector comprised of the coordinates of x in the basis $f^1, ..., f^m$, meets all the requirements.

Proof of (iii) is given by important by its own right *Gram-Schmidt orthogonalization process* as follows. We start with an arbitrary basis $h^1, ..., h^m$ in L and step by step convert it into an orthonormal basis $f^1, ..., f^m$. At the beginning of a step t of the construction, we already have an orthonormal collection $f^1, ..., f^{t-1}$ such that $\mathrm{Lin}\{f^1, ..., f^{t-1}\} = \mathrm{Lin}\{h^1, ..., h^{t-1}\}$. At a step t we

1. Build the vector
$$g^t = h^t - \sum_{j=1}^{t-1} \langle h^t, f^j\rangle f^j.$$
It is easily seen (check it!) that
   (a) One has
$$\mathrm{Lin}\{f^1, ..., f^{t-1}, g^t\} = \mathrm{Lin}\{h^1, ..., h^t\}; \tag{A.2.1}$$
   (b) $g^t \neq 0$ (derive this fact from (A.2.1) and the linear independence of the collection $h^1, ..., h^m$);
   (c) g^t is orthogonal to $f^1, ..., f^{t-1}$
2. Since $g^t \neq 0$, the quantity $\langle g^t, g^t\rangle$ is positive (positive definiteness of the inner product), so that the vector
$$f^t = \frac{1}{\sqrt{\langle g^t, g^t\rangle}} g^t$$
is well defined. It is immediately seen (check it!) that the collection $f^1, ..., f^t$ is orthonormal and
$$\mathrm{Lin}\{f^1, ..., f^t\} = \mathrm{Lin}\{f^1, ..., f^{t-1}, g^t\} = \mathrm{Lin}\{h^1, ..., h^t\}.$$
Step t of the orthogonalization process is completed.

After m steps of the optimization process, we end up with an orthonormal system $f^1, ..., f^m$ of vectors from L such that

$$\mathrm{Lin}\{f^1, ..., f^m\} = \mathrm{Lin}\{h^1, ..., h^m\} = L,$$

so that $f^1, ..., f^m$ is an orthonormal basis in L.

The construction can be easily modified (do it!) to extend a given orthonormal system of vectors from L to an orthonormal basis of L.

Exercise A.2 *1. What is the orthogonal complement (w.r.t. the standard inner product) of the subspace* $\{x \in \mathbf{R}^n : \sum_{i=1}^n x_i = 0\}$ *in* $\mathbf{R}^n$*?*

2. Find an orthonormal basis (w.r.t. the standard inner product) in the linear subspace $\{x \in \mathbf{R}^n : x_1 = 0\}$ *of* $\mathbf{R}^n$

3. Let L *be a linear subspace of* $\mathbf{R}^n$*, and* $f^1, ..., f^m$ *be an orthonormal basis in* L*. Prove that for every* $x \in \mathbf{R}^n$*, the orthogonal projection* x_L *of* x *onto* L *is given by the formula*

$$x_L = \sum_{i=1}^m (x^T f^i) f^i.$$

4. Let L_1, L_2 *be linear subspaces in* $\mathbf{R}^n$*. Verify the formulas*

$$(L_1 + L_2)^\perp = L_1^\perp \cap L_2^\perp; \quad (L_1 \cap L_2)^\perp = L_1^\perp + L_2^\perp.$$

5. Consider the space of $m \times n$ *matrices* $\mathbf{R}^{m\times n}$*, and let us equip it with the "standard inner product" (called the Frobenius inner product)*

$$\langle A, B\rangle = \sum_{i,j} A_{ij} B_{ij}$$

(as if we were treating $m\times n$ *matrices as* mn*-dimensional vectors, writing the entries of the matrices column by column, and then taking the standard inner product of the resulting long vectors).*

(a) Verify that in terms of matrix multiplication the Frobenius inner product can be written as

$$\langle A, B\rangle = \mathrm{Tr}(AB^T),$$

where $\mathrm{Tr}(C)$ *is the trace (the sum of diagonal elements) of a square matrix* C*.*

(b) Build an orthonormal basis in the linear subspace $\mathbf{S}^n$ *of symmetric* $n \times n$ *matrices*

(c) What is the orthogonal complement of the subspace $\mathbf{S}^n$ *of symmetric* $n \times n$ *matrices in the space* $\mathbf{R}^{n\times n}$ *of square* $n \times n$ *matrices?*

(d) Find the orthogonal decomposition, w.r.t. $\mathbf{S}^2$*, of the matrix* $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

A.3 Affine subspaces in $\mathbf{R}^n$

Many of events to come will take place not in the entire $\mathbf{R}^n$, but in its *affine subspaces* which, geometrically, are planes of different dimensions in $\mathbf{R}^n$. Let us become acquainted with these subspaces.

A.3.1 Affine subspaces and affine hulls

Definition of an affine subspace. Geometrically, a linear subspace L of $\mathbf{R}^n$ is a special plane – the one passing through the origin of the space (i.e., containing the zero vector). To get an arbitrary plane M, it suffices to subject an appropriate special plane L to a translation – to add to all points from L a fixed *shifting vector* a. This geometric intuition leads to the following

Definition A.3.1 [Affine subspace] *An affine subspace (a plane) in* $\mathbf{R}^n$ *is a set of the form*

$$M = a + L = \{y = a + x : x \in L\}, \tag{A.3.1}$$

where L *is a linear subspace in* $\mathbf{R}^n$ *and* a *is a vector from* $\mathbf{R}^n$ [2)].

[2)] according to our convention on arithmetic of sets, I was supposed to write in (A.3.1) $\{a\} + L$ instead of $a + L$ – we did not define arithmetic sum of a vector and a set. Usually people ignore this difference and omit the brackets when writing down singleton sets in similar expressions: we shall write $a + L$ instead of $\{a\} + L$, $\mathbf{R}d$ instead of $\mathbf{R}\{d\}$, etc.

E.g., shifting the linear subspace L comprised of vectors with zero first entry by a vector $a = (a_1, ..., a_n)$, we get the set $M = a + L$ of all vectors x with $x_1 = a_1$; according to our terminology, this is an affine subspace.

Immediate question about the notion of an affine subspace is: what are the "degrees of freedom" in decomposition (A.3.1) – how "strict" M determines a and L? The answer is as follows:

Proposition A.3.1 *The linear subspace L in decomposition (A.3.1) is uniquely defined by M and is the set of all differences of the vectors from M:*

$$L = M - M = \{x - y : x, y \in M\}. \tag{A.3.2}$$

The shifting vector a is not uniquely defined by M and can be chosen as an arbitrary vector from M.

A.3.2 Intersections of affine subspaces, affine combinations and affine hulls

An immediate conclusion of Proposition A.3.1 is as follows:

Corollary A.3.1 *Let $\{M_\alpha\}$ be an arbitrary family of affine subspaces in $\mathbf{R}^n$, and assume that the set $M = \cap_\alpha M_\alpha$ is nonempty. Then M_α is an affine subspace.*

From Corollary A.3.1 it immediately follows that for every nonempty subset Y of $\mathbf{R}^n$ there exists the smallest affine subspace containing Y – the intersection of all affine subspaces containing Y. This smallest affine subspace containing Y is called the *affine hull* of Y (notation: $\mathrm{Aff}(Y)$).

All this resembles a lot the story about linear spans. Can we further extend this analogy and to get a description of the affine hull $\mathrm{Aff}(Y)$ in terms of elements of Y similar to the one of the linear span ("linear span of X is the set of all linear combinations of vectors from X")? Sure we can!

Let us choose somehow a point $y_0 \in Y$, and consider the set

$$X = Y - y_0.$$

All affine subspaces containing Y should contain also y_0 and therefore, by Proposition A.3.1, can be represented as $M = y_0 + L$, L being a linear subspace. It is absolutely evident that an affine subspace $M = y_0 + L$ contains Y if and only if the subspace L contains X, and that the larger is L, the larger is M:

$$L \subset L' \Rightarrow M = y_0 + L \subset M' = y_0 + L'.$$

Thus, to find the smallest among *affine subspaces containing* Y, it suffices to find the smallest among the *linear subspaces containing* X and to translate the latter space by y_0:

$$\mathrm{Aff}(Y) = y_0 + \mathrm{Lin}(X) = y_0 + \mathrm{Lin}(Y - y_0). \tag{A.3.3}$$

Now, we know what is $\mathrm{Lin}(Y - y_0)$ – this is a set of all linear combinations of vectors from $Y - y_0$, so that a generic element of $\mathrm{Lin}(Y - y_0)$ is

$$x = \sum_{i=1}^{k} \mu_i (y_i - y_0) \quad [k \text{ may depend of } x]$$

with $y_i \in Y$ and real coefficients μ_i. It follows that the generic element of $\mathrm{Aff}(Y)$ is

$$y = y_0 + \sum_{i=1}^{k} \mu_i (y_i - y_0) = \sum_{i=0}^{k} \lambda_i y_i,$$

where

$$\lambda_0 = 1 - \sum_i \mu_i, \ \lambda_i = \mu_i, \ i \geq 1.$$

We see that a generic element of $\mathrm{Aff}(Y)$ is a linear combination of vectors from Y. Note, however, that the coefficients λ_i in this combination are not completely arbitrary: their sum is equal to 1. Linear combinations of this type – with the unit sum of coefficients – have a special name – they are called affine combinations.

We have seen that every vector from $\mathrm{Aff}(Y)$ is an affine combination of vectors of Y. Whether the inverse is true, i.e., whether $\mathrm{Aff}(Y)$ contains all affine combinations of vectors from Y? The answer is positive. Indeed, if

$$y = \sum_{i=1}^{k} \lambda_i y_i$$

is an affine combination of vectors from Y, then, using the equality $\sum_i \lambda_i = 1$, we can write it also as

$$y = y_0 + \sum_{i=1}^{k} \lambda_i (y_i - y_0),$$

y_0 being the "marked" vector we used in our previous reasoning, and the vector of this form, as we already know, belongs to $\mathrm{Aff}(Y)$. Thus, we come to the following

Proposition A.3.2 [Structure of affine hull]

$$\mathrm{Aff}(Y) = \{\textit{the set of all affine combinations of vectors from } Y\}.$$

When Y itself is an affine subspace, it, of course, coincides with its affine hull, and the above Proposition leads to the following

Corollary A.3.2 *An affine subspace M is closed with respect to taking affine combinations of its members – every combination of this type is a vector from M. Vice versa, a nonempty set which is closed with respect to taking affine combinations of its members is an affine subspace.*

A.3.3 Affinely spanning sets, affinely independent sets, affine dimension

Affine subspaces are closely related to linear subspaces, and the basic notions associated with linear subspaces have natural and useful affine analogies. Here we introduce these notions and discuss their basic properties.

Affinely spanning sets. Let $M = a + L$ be an affine subspace. We say that a subset Y of M is *affinely spanning* for M (we say also that Y spans M affinely, or that M is affinely spanned by Y), if $M = \mathrm{Aff}(Y)$, or, which is the same due to Proposition A.3.2, if every point of M is an affine combination of points from Y. An immediate consequence of the reasoning of the previous Section is as follows:

Proposition A.3.3 *Let $M = a + L$ be an affine subspace and Y be a subset of M, and let $y_0 \in Y$. The set Y affinely spans M – $M = \mathrm{Aff}(Y)$ – if and only if the set*

$$X = Y - y_0$$

spans the linear subspace L: $L = \mathrm{Lin}(X)$.

Affinely independent sets. A linearly independent set $x_1, ..., x_k$ is a set such that no nontrivial linear combination of $x_1, ..., x_k$ equals to zero. An equivalent definition is given by Theorem A.1.2.(iv): $x_1, ..., x_k$ are linearly independent, if the coefficients in a linear combination

$$x = \sum_{i=1}^{k} \lambda_i x_i$$

are uniquely defined by the value x of the combination. This equivalent form reflects the essence of the matter – what we indeed need, is the uniqueness of the coefficients in expansions. Accordingly, this equivalent form is the prototype for the notion of an affinely independent set: we want to introduce this notion in such a way that the coefficients λ_i in an *affine* combination

$$y = \sum_{i=0}^{k} \lambda_i y_i$$

of "affinely independent" set of vectors $y_0, ..., y_k$ would be uniquely defined by y. *Non*-uniqueness would mean that

$$\sum_{i=0}^{k} \lambda_i y_i = \sum_{i=0}^{k} \lambda'_i y_i$$

for two different collections of coefficients λ_i and λ'_i with unit sums of coefficients; if it is the case, then

$$\sum_{i=0}^{m} (\lambda_i - \lambda'_i) y_i = 0,$$

so that y_i's are linearly dependent and, moreover, there exists a nontrivial zero combination of then with *zero sum of coefficients* (since $\sum\limits_i (\lambda_i - \lambda'_i) = \sum\limits_i \lambda_i - \sum\limits_i \lambda'_i = 1 - 1 = 0$). Our reasoning can be inverted – if there exists a nontrivial linear combination of y_i's with zero sum of coefficients which is zero, then the coefficients in the representation of a vector as an affine combination of y_i's are not uniquely defined. Thus, in order to get uniqueness we should for sure forbid relations

$$\sum_{i=0}^{k} \mu_i y_i = 0$$

with nontrivial zero sum coefficients μ_i. Thus, we have motivated the following

Definition A.3.2 [Affinely independent set] *A collection $y_0, ..., y_k$ of n-dimensional vectors is called affinely independent, if no nontrivial linear combination of the vectors with zero sum of coefficients is zero:*

$$\sum_{i=1}^{k} \lambda_i y_i = 0, \ \sum_{i=0}^{k} \lambda_i = 0 \Rightarrow \lambda_0 = \lambda_1 = ... = \lambda_k = 0.$$

With this definition, we get the result completely similar to the one of Theorem A.1.2.(iv):

Corollary A.3.3 *Let $y_0, ..., y_k$ be affinely independent. Then the coefficients λ_i in an affine combination*

$$y = \sum_{i=0}^{k} \lambda_i y_i \quad [\sum_i \lambda_i = 1]$$

of the vectors $y_0, ..., y_k$ are uniquely defined by the value y of the combination.

Verification of affine independence of a collection can be immediately reduced to verification of linear independence of closely related collection:

Proposition A.3.4 *$k+1$ vectors $y_0, ..., y_k$ are affinely independent if and only if the k vectors $(y_1 - y_0), (y_2 - y_0), ..., (y_k - y_0)$ are linearly independent.*

From the latter Proposition it follows, e.g., that the collection $0, e_1, ..., e_n$ comprised of the origin and the standard basic orths is affinely independent. Note that this collection is linearly dependent (as every collection containing zero). You should definitely know the difference between the two notions of independence we deal with: linear independence means that no nontrivial linear combination of the vectors can be zero, while affine independence means that no nontrivial linear combination *from certain restricted class of them* (with zero sum of coefficients) can be zero. Therefore, there are more affinely independent sets than the linearly independent ones: a linearly independent set is for sure affinely independent, but not vice versa.

Affine bases and affine dimension. Propositions A.3.2 and A.3.3 reduce the notions of affine spanning/affine independent sets to the notions of spanning/linearly independent ones. Combined with Theorem A.1.2, they result in the following analogies of the latter two statements:

Proposition A.3.5 [Affine dimension] *Let $M = a + L$ be an affine subspace in $\mathbf{R}^n$. Then the following two quantities are finite integers which are equal to each other:*

(i) *minimal # of elements in the subsets of M which affinely span M;*

(ii) *maximal # of elements in affine independent subsets of M.*

The common value of these two integers is by 1 more than the dimension dim L *of L.*

By definition, the *affine dimension* of an affine subspace $M = a+L$ is the dimension dim L of L. Thus, if M is of affine dimension k, then the minimal cardinality of sets affinely spanning M, same as the maximal cardinality of affine independent subsets of M, is $k + 1$.

Theorem A.3.1 [Affine bases] *Let $M = a + L$ be an affine subspace in $\mathbf{R}^n$.*

A. *Let $Y \subset M$. The following three properties of X are equivalent:*

(i) *Y is an affine independent set which affinely spans M;*

(ii) *Y is affine independent and contains $1 +$* dim L *elements;*

(iii) *Y affinely spans M and contains $1 +$* dim L *elements.*

A subset Y of M possessing the indicated equivalent to each other properties is called an affine basis of M. Affine bases in M are exactly the collections $y_0, ..., y_{\dim L}$ such that $y_0 \in M$ and $(y_1 - y_0), ..., (y_{\dim L} - y_0)$ is a basis in L.

B. *Every affinely independent collection of vectors of M either itself is an affine basis of M, or can be extended to such a basis by adding new vectors. In particular, there exists affine basis of M.*

C. *Given a set Y which affinely spans M, you can always extract from this set an affine basis of M.*

We already know that the standard basic orths $e_1, ..., e_n$ form a basis of the entire space $\mathbf{R}^n$. And what about affine bases in $\mathbf{R}^n$? According to Theorem A.3.1.**A**, you can choose as such a basis a collection $e_0, e_0 + e_1, ..., e_0 + e_n$, e_0 being an arbitrary vector.

Barycentric coordinates. Let M be an affine subspace, and let $y_0, ..., y_k$ be an affine basis of M. Since the basis, by definition, affinely spans M, every vector y from M is an affine combination of the vectors of the basis:

$$y = \sum_{i=0}^{k} \lambda_i y_i \quad [\sum_{i=0}^{k} \lambda_i = 1],$$

and since the vectors of the affine basis are affinely independent, the coefficients of this combination are uniquely defined by y (Corollary A.3.3). These coefficients are called *barycentric coordinates* of y with respect to the affine basis in question. In contrast to the usual coordinates with respect to a (linear) basis, the barycentric coordinates could not be quite arbitrary: their sum should be equal to 1.

A.3.4 Dual description of linear subspaces and affine subspaces

To the moment we have introduced the notions of linear subspace and affine subspace and have presented a scheme of generating these entities: to get, e.g., a linear subspace, you start from an arbitrary nonempty set $X \subset \mathbf{R}^n$ and add to it all linear combinations of the vectors from X. When replacing linear combinations with the affine ones, you get a way to generate affine subspaces.

The just indicated way of generating linear subspaces/affine subspaces resembles the approach of a worker building a house: he starts with the base and then adds to it new elements until the house is ready. There exists, anyhow, an approach of an artist creating a sculpture: he takes something large and then deletes extra parts of it. Is there something like "artist's way" to represent linear subspaces and affine subspaces? The answer is positive and very instructive.

A.3.4.1 Affine subspaces and systems of linear equations

Let L be a linear subspace. According to Theorem A.2.2.(i), it is an orthogonal complement – namely, the orthogonal complement to the linear subspace $L^\perp$. Now let $a_1, ..., a_m$ be a finite spanning set in $L^\perp$. A vector x which is orthogonal to $a_1, ..., a_m$ is orthogonal to the entire $L^\perp$ (since every vector from $L^\perp$ is a linear combination of $a_1, ..., a_m$ and the inner product is bilinear); and of course vice versa, a vector orthogonal to the entire $L^\perp$ is orthogonal to $a_1, ..., a_m$. We see that

$$L = (L^\perp)^\perp = \{x : a_i^T x = 0,\ i = 1, ..., k\}. \tag{A.3.4}$$

Thus, we get a very important, although simple,

Proposition A.3.6 ["Outer" description of a linear subspace] *Every linear subspace L in $\mathbf{R}^n$ is a set of solutions to a homogeneous linear system of equations*

$$a_i^T x = 0,\ i = 1, ..., m, \tag{A.3.5}$$

given by properly chosen m and vectors $a_1, ..., a_m$.

Proposition A.3.6 is an "if and only if" statement: as we remember from Example A.1.3.A.4, solution set to a homogeneous system of linear equations with n variables always is a linear subspace in $\mathbf{R}^n$.

From Proposition A.3.6 and the facts we know about the dimension we can easily derive several important consequences:

- Systems (A.3.5) which define a given linear subspace L are exactly the systems given by the vectors $a_1, ..., a_m$ which span $L^\perp$ [3)]
- The smallest possible number m of equations in (A.3.5) is the dimension of $L^\perp$, i.e., by Remark A.2.2, is codim $L \equiv n - \dim L$ [4)]

Now, an affine subspace M is, by definition, a translation of a linear subspace: $M = a + L$. As we know, vectors x from L are exactly the solutions of certain *homogeneous* system of linear equations

$$a_i^T x = 0,\ i = 1, ..., m.$$

It is absolutely clear that adding to these vectors a fixed vector a, we get exactly the set of solutions to the *inhomogeneous* solvable linear system

$$a_i^T x = b_i \equiv a_i^T a,\ i = 1, ..., m.$$

Vice versa, the set of solutions to a *solvable* system of linear equations

$$a_i^T x = b_i,\ i = 1, ..., m,$$

with n variables is the sum of a particular solution to the system and the solution set to the corresponding homogeneous system (the latter set, as we already know, is a linear subspace in $\mathbf{R}^n$), i.e., is an affine subspace. Thus, we get the following

Proposition A.3.7 ["Outer" description of an affine subspace]
Every affine subspace $M = a + L$ in $\mathbf{R}^n$ is a set of solutions to a solvable linear system of equations

$$a_i^T x = b_i,\ i = 1, ..., m, \tag{A.3.6}$$

given by properly chosen m and vectors $a_1, ..., a_m$.

Vice versa, the set of all solutions to a solvable system of linear equations with n variables is an affine subspace in $\mathbf{R}^n$.

The linear subspace L associated with M is exactly the set of solutions of the homogeneous (with the right hand side set to 0) version of system (A.3.6).

We see, in particular, that an affine subspace always is closed.

[3)] the reasoning which led us to Proposition A.3.6 says that [$a_1, ..., a_m$ span $L^\perp$] $\Rightarrow$ [(A.3.5) defines L]; now we claim that the inverse also is true

[4)] to make this statement true also in the extreme case when $L = \mathbf{R}^n$ (i.e., when codim $L = 0$), we from now on make a convention that an *empty* set of equations or inequalities defines, as the solution set, the entire space

Comment. The "outer" description of a linear subspace/affine subspace – the "artist's" one – is in many cases much more useful than the "inner" description via linear/affine combinations (the "worker's" one). E.g., with the outer description it is very easy to check whether a given vector belongs or does not belong to a given linear subspace/affine subspace, which is not that easy with the inner one[5)]. In fact both descriptions are "complementary" to each other and perfectly well work in parallel: what is difficult to see with one of them, is clear with another. The idea of using "inner" and "outer" descriptions of the entities we meet with – linear subspaces, affine subspaces, convex sets, optimization problems – the general idea of *duality* – is, I would say, the main driving force of Convex Analysis and Optimization, and in the sequel we would all the time meet with different implementations of this fundamental idea.

A.3.5 Structure of the simplest affine subspaces

This small subsection deals mainly with terminology. According to their dimension, affine subspaces in $\mathbf{R}^n$ are named as follows:

- Subspaces of dimension 0 are translations of the only 0-dimensional linear subspace $\{0\}$, i.e., are singleton sets – vectors from $\mathbf{R}^n$. These subspaces are called *points*; a point is a solution to a square system of linear equations with nonsingular matrix.

- Subspaces of dimension 1 (lines). These subspaces are translations of one-dimensional linear subspaces of $\mathbf{R}^n$. A one-dimensional linear subspace has a single-element basis given by a nonzero vector d and is comprised of all multiples of this vector. Consequently, line is a set of the form

  $$\{y = a + td : t \in \mathbf{R}\}$$

  given by a pair of vectors a (the origin of the line) and d (the direction of the line), $d \neq 0$. The origin of the line and its direction are not uniquely defined by the line; you can choose as origin any point on the line and multiply a particular direction by nonzero reals.

  In the barycentric coordinates a line is described as follows:

  $$l = \{\lambda_0 y_0 + \lambda_1 y_1 : \lambda_0 + \lambda_1 = 1\} = \{\lambda y_0 + (1-\lambda) y_1 : \lambda \in \mathbf{R}\},$$

  where y_0, y_1 is an affine basis of l; you can choose as such a basis any pair of distinct points on the line.

  The "outer" description a line is as follows: it is the set of solutions to a linear system with n variables and $n-1$ linearly independent equations.

- Subspaces of dimension > 2 and $< n-1$ have no special names; sometimes they are called affine planes of such and such dimension.

- Affine subspaces of dimension $n-1$, due to important role they play in Convex Analysis, have a special name – they are called *hyperplanes*. The outer description of a hyperplane is that a hyperplane is the solution set of a *single* linear equation

  $$a^T x = b$$

  with nontrivial left hand side ($a \neq 0$). In other words, a hyperplane is the level set $a(x) = \text{const}$ of a nonconstant linear form $a(x) = a^T x$.

- The "largest possible" affine subspace – the one of dimension n – is unique and is the entire $\mathbf{R}^n$. This subspace is given by an empty system of linear equations.

[5)] in principle it is not difficult to certify that a given point belongs to, say, a linear subspace given as the linear span of some set – it suffices to point out a representation of the point as a linear combination of vectors from the set. But how could you certify that the point does *not* belong to the subspace?

A.4 Space $\mathbf{R}^n$: metric structure and topology

Euclidean structure on the space $\mathbf{R}^n$ gives rise to a number of extremely important *metric* notions – distances, convergence, etc. For the sake of definiteness, we associate these notions with the standard inner product $x^T y$.

A.4.1 Euclidean norm and distances

By positive definiteness, the quantity $x^T x$ always is nonnegative, so that the quantity

$$|x| \equiv \| x \|_2 = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$$

is well-defined; this quantity is called the (standard) *Euclidean norm* of vector x (or simply the norm of x) and is treated as the distance from the origin to x. The distance between two arbitrary points $x, y \in \mathbf{R}^n$ is, by definition, the norm $|x - y|$ of the difference $x - y$. The notions we have defined satisfy all basic requirements on the general notions of a norm and distance, specifically:

1. *Positivity of norm: The norm of a vector always is nonnegative; it is zero if and only is the vector is zero*:
$$|x| \geq 0 \quad \forall x; \quad |x| = 0 \Leftrightarrow x = 0.$$
2. *Homogeneity of norm: When a vector is multiplied by a real, its norm is multiplied by the absolute value of the real*:
$$|\lambda x| = |\lambda| \cdot |x| \quad \forall (x \in \mathbf{R}^n, \lambda \in \mathbf{R}).$$
3. *Triangle inequality: Norm of the sum of two vectors is $\leq$ the sum of their norms*:
$$|x + y| \leq |x| + |y| \quad \forall (x, y \in \mathbf{R}^n).$$

   In contrast to the properties of positivity and homogeneity, which are absolutely evident, the Triangle inequality is not trivial and definitely requires a proof. The proof goes through a fact which is extremely important by its own right – the *Cauchy Inequality*, which perhaps is the most frequently used inequality in Mathematics:

   Theorem A.4.1 [Cauchy's Inequality] *The absolute value of the inner product of two vectors does not exceed the product of their norms:*

   $$|x^T y| \leq |x||y| \quad \forall (x, y \in \mathbf{R}^n)$$

   and is equal to the product of the norms if and only if one of the vectors is proportional to the other one:

   $$|x^T y| = |x||y| \Leftrightarrow \{\exists \alpha : x = \alpha y \text{ or } \exists \beta : y = \beta x\}$$

   Proof is immediate: we may assume that both x and y are nonzero (otherwise the Cauchy inequality clearly is equality, and one of the vectors is constant times (specifically, zero times) the other one, as announced in Theorem). Assuming $x, y \neq 0$, consider the function
   $$f(\lambda) = (x - \lambda y)^T (x - \lambda y) = x^T x - 2\lambda x^T y + \lambda^2 y^T y.$$
   By positive definiteness of the inner product, this function – which is a second order polynomial – is nonnegative on the entire axis, whence the discriminant of the polynomial
   $$(x^T y)^2 - (x^T x)(y^T y)$$
   is nonpositive:
   $$(x^T y)^2 \leq (x^T x)(y^T y).$$

Taking square roots of both sides, we arrive at the Cauchy Inequality. We also see that the inequality is equality if and only if the discriminant of the second order polynomial $f(\lambda)$ is zero, i.e., if and only if the polynomial has a (multiple) real root; but due to positive definiteness of inner product, $f(\cdot)$ has a root λ if and only if $x = \lambda y$, which proves the second part of Theorem. ∎

From Cauchy's Inequality to the Triangle Inequality: Let $x, y \in \mathbf{R}^n$. Then

$$\begin{array}{rcll} |x+y|^2 & = & (x+y)^T(x+y) & \text{[definition of norm]} \\ & = & x^Tx + y^Ty + 2x^Ty & \text{[opening parentheses]} \\ & \leq & \underbrace{x^Tx}_{|x|^2} + \underbrace{y^Ty}_{|y|^2} + 2|x||y| & \text{[Cauchy's Inequality]} \\ & = & (|x| + |y|)^2 & \\ \Rightarrow |x+y| & \leq & |x| + |y| & \end{array}$$ ∎

The properties of norm (i.e., of the distance to the origin) we have established induce properties of the distances between pairs of arbitrary points in $\mathbf{R}^n$, specifically:

1. *Positivity of distances: The distance $|x-y|$ between two points is positive, except for the case when the points coincide ($x = y$), when the distance between x and y is zero;*
2. *Symmetry of distances: The distance from x to y is the same as the distance from y to x:*
$$|x - y| = |y - x|;$$
3. *Triangle inequality for distances:* For every three points x, y, z, the distance from x to z does not exceed the sum of distances between x and y and between y and z:
$$|z - x| \leq |y - x| + |z - y| \quad \forall (x, y, z \in \mathbf{R}^n)$$

A.4.2 Convergence

Equipped with distances, we can define the fundamental notion of *convergence of a sequence of vectors*. Specifically, we say that *a sequence $x^1, x^2, ...$ of vectors from $\mathbf{R}^n$ converges to a vector $\bar{x}$*, or, equivalently, that *$\bar{x}$ is the limit of the sequence $\{x^i\}$* (notation: $\bar{x} = \lim\limits_{i\to\infty} x^i$), if the distances from $\bar{x}$ to x^i go to 0 as $i \to \infty$:

$$\bar{x} = \lim_{i\to\infty} x^i \Leftrightarrow |\bar{x} - x^i| \to 0, i \to \infty,$$

or, which is the same, for every $\epsilon > 0$ there exists $i = i(\epsilon)$ such that the distance between every point x^i, $i \geq i(\epsilon)$, and $\bar{x}$ does not exceed ϵ:

$$\left\{|\bar{x} - x^i| \to 0, i \to \infty\right\} \Leftrightarrow \left\{\forall \epsilon > 0 \exists i(\epsilon) : i \geq i(\epsilon) \Rightarrow |\bar{x} - x^i| \leq \epsilon\right\}.$$

Exercise A.3 *Verify the following facts:*

1. *$\bar{x} = \lim\limits_{i\to\infty} x^i$ if and only if for every $j = 1, ..., n$ the coordinates # j of the vectors x^i converge, as $i \to \infty$, to the coordinate # j of the vector $\bar{x}$;*
2. *If a sequence converges, its limit is uniquely defined;*
3. *Convergence is compatible with linear operations:*
   — *if $x^i \to x$ and $y^i \to y$ as $i \to \infty$, then $x^i + y^i \to x + y$ as $i \to \infty$;*
   — *if $x^i \to x$ and $\lambda_i \to \lambda$ as $i \to \infty$, then $\lambda_i x^i \to \lambda x$ as $i \to \infty$.*

A.4.3 Closed and open sets

After we have in our disposal distance and convergence, we can speak about *closed* and *open* sets:

- A set $X \subset \mathbf{R}^n$ is called *closed*, if it contains limits of all converging sequences of elements of X:

$$\left\{x^i \in X, x = \lim_{i\to\infty} x^i\right\} \Rightarrow x \in X$$

- A set $X \subset \mathbf{R}^n$ is called *open*, if whenever x belongs to X, all points close enough to x also belong to X:

$$\forall (x \in X)\exists(\delta > 0) : |x' - x| < \delta \Rightarrow x' \in X.$$

An open set containing a point x is called a *neighbourhood* of x.

Examples of closed sets: (1) $\mathbf{R}^n$; (2) $\emptyset$; (3) the sequence $x^i = (i, 0, ..., 0)$, $i = 1, 2, 3, ...$; (4) $\{x \in \mathbf{R}^n : \sum_{i=1}^{n} a_{ij}x_j = 0,\ i = 1, ..., m\}$ (in other words: a linear subspace in $\mathbf{R}^n$ always is closed, see Proposition A.3.6);(5) $\{x \in \mathbf{R}^n : \sum_{i=1}^{n} a_{ij}x_j = b_i,\ i = 1, ..., m\}$ (in other words: an affine subset of $\mathbf{R}^n$ always is closed, see Proposition A.3.7);; (6) Any finite subset of $\mathbf{R}^n$

Examples of non-closed sets: (1) $\mathbf{R}^n \backslash \{0\}$; (2) the sequence $x^i = (1/i, 0, ..., 0)$, $i = 1, 2, 3, ...$; (3) $\{x \in \mathbf{R}^n : x_j > 0, j = 1, ..., n\}$; (4) $\{x \in \mathbf{R}^n : \sum_{i=1}^{n} x_j > 5\}$.

Examples of open sets: (1) $\mathbf{R}^n$; (2) $\emptyset$; (3) $\{x \in \mathbf{R}^n : \sum_{j=1}^{n} a_{ij}x_j > b_j, i = 1, ..., m\}$; (4) complement of a finite set.

Examples of non-open sets: (1) A nonempty finite set; (2) the sequence $x^i = (1/i, 0, ..., 0)$, $i = 1, 2, 3, ...$, and the sequence $x^i = (i, 0, 0, ..., 0)$, $i = 1, 2, 3, ...$; (3) $\{x \in \mathbf{R}^n : x_j \geq 0, j = 1, ..., n\}$; (4) $\{x \in \mathbf{R}^n : \sum_{i=1}^{n} x_j \geq 5\}$.

Exercise A.4 *Mark in the list to follows those sets which are closed and those which are open:*

1. *All vectors with integer coordinates*
2. *All vectors with rational coordinates*
3. *All vectors with positive coordinates*
4. *All vectors with nonnegative coordinates*
5. $\{x : |x| < 1\}$*;*
6. $\{x : |x| = 1\}$*;*
7. $\{x : |x| \leq 1\}$*;*
8. $\{x : |x| \geq 1\}$*:*
9. $\{x : |x| > 1\}$*;*
10. $\{x : 1 < |x| \leq 2\}$.

Verify the following facts

1. *A set $X \subset \mathbf{R}^n$ is closed if and only if its complement $\bar{X} = \mathbf{R}^n \backslash X$ is open;*
2. *Intersection of every family (finite or infinite) of closed sets is closed. Union of every family (finite of infinite) of open sets is open.*
3. *Union of finitely many closed sets is closed. Intersection of finitely many open sets is open.*

A.4.4 Local compactness of $\mathbf{R}^n$

A fundamental fact about convergence in $\mathbf{R}^n$, which in certain sense is characteristic for this series of spaces, is the following

Theorem A.4.2 *From every bounded sequence $\{x^i\}_{i=1}^{\infty}$ of points from $\mathbf{R}^n$ one can extract a converging subsequence $\{x^{i_j}\}_{j=1}^{\infty}$. Equivalently: A closed and bounded subset X of $\mathbf{R}^n$ is compact, i.e., a set possessing the following two equivalent to each other properties:*

(i) *From every sequence of elements of X one can extract a subsequence which converges to certain point of X;*

(ii) *From every open covering of X (i.e., a family $\{U_\alpha\}_{\alpha\in A}$ of open sets such that $X \subset \bigcup\limits_{\alpha\in A} U_\alpha$) one can extract a finite sub-covering, i.e., a finite subset of indices $\alpha_1, ..., \alpha_N$ such that $X \subset \bigcup\limits_{i=1}^{N} U_{\alpha_i}$.*

A.5 Continuous functions on $\mathbf{R}^n$

A.5.1 Continuity of a function

Let $X \subset \mathbf{R}^n$ and $f(x) : X \to \mathbf{R}^m$ be a function (another name – mapping) defined on X and taking values in $\mathbf{R}^m$.

1. f is called *continuous at a point* $\bar{x} \in X$, if for every sequence x^i of points of X converging to $\bar{x}$ the sequence $f(x^i)$ converges to $f(\bar{x})$. Equivalent definition:

   $f : X \to \mathbf{R}^m$ is continuous at $\bar{x} \in X$, if for every $\epsilon > 0$ there exists $\delta > 0$ such that

   $$x \in X, |x - \bar{x}| < \delta \Rightarrow |f(x) - f(\bar{x})| < \epsilon.$$

2. f is called *continuous on* X, if f is continuous at every point from X. Equivalent definition: f preserves convergence: whenever a sequence of points $x^i \in X$ converges to a point $x \in X$, the sequence $f(x^i)$ converges to $f(x)$.

   Examples of continuous mappings:

   1. An *affine* mapping

      $$f(x) = \begin{bmatrix} \sum\limits_{j=1}^{m} A_{1j}x_j + b_1 \\ \vdots \\ \sum\limits_{j=1}^{m} A_{mj}x_j + b_m \end{bmatrix} \equiv Ax + b : \mathbf{R}^n \to \mathbf{R}^m$$

      is continuous on the entire $\mathbf{R}^n$ (and thus – on every subset of $\mathbf{R}^n$) (check it!).
   2. The norm $|x|$ is a continuous on $\mathbf{R}^n$ (and thus – on every subset of $\mathbf{R}^n$) real-valued function (check it!).

Exercise A.5 • *Consider the function*

$$f(x_1, x_2) = \begin{cases} \frac{x_1^2 - x_2^2}{x_1^2 + x_2^2}, & (x_1, x_2) \neq 0 \\ 0, & x_1 = x_2 = 0 \end{cases} : \mathbf{R}^2 \to \mathbf{R}.$$

Check whether this function is continuous on the following sets:

1. $\mathbf{R}^2$;
2. $\mathbf{R}^2 \backslash \{0\}$;

3. $\{x \in \mathbf{R}^2 : x_1 = 0\}$;
4. $\{x \in \mathbf{R}^2 : x_2 = 0\}$;
5. $\{x \in \mathbf{R}^2 : x_1 + x_2 = 0\}$;
6. $\{x \in \mathbf{R}^2 : x_1 - x_2 = 0\}$;
7. $\{x \in \mathbf{R}^2 : |x_1 - x_2| \le x_1^4 + x_2^4\}$;

- *Let $f : \mathbf{R}^n \to \mathbf{R}^m$ be a continuous mapping. Mark those of the following statements which always are true:*

1. *If U is an open set in $\mathbf{R}^m$, then so is the set $f^{-1}(U) = \{x : f(x) \in U\}$;*
2. *If U is an open set in $\mathbf{R}^n$, then so is the set $f(U) = \{f(x) : x \in U\}$;*
3. *If F is a closed set in $\mathbf{R}^m$, then so is the set $f^{-1}(F) = \{x : f(x) \in F\}$;*
4. *If F is an closed set in $\mathbf{R}^n$, then so is the set $f(F) = \{f(x) : x \in F\}$.*

A.5.2 Elementary continuity-preserving operations

All "elementary" operations with mappings preserve continuity. Specifically,

Theorem A.5.1 *Let X be a subset in $\mathbf{R}^n$.*
(i) [stability of continuity w.r.t. linear operations] *If $f_1(x), f_2(x)$ are continuous functions on X taking values in $\mathbf{R}^m$ and $\lambda_1(x)$, $\lambda_2(x)$ are continuous real-valued functions on X, then the function*

$$f(x) = \lambda_1(x)f_1(x) + \lambda_2(x)f_2(x) : X \to \mathbf{R}^m$$

is continuous on X;
(ii) [stability of continuity w.r.t. superposition] *Let*

- *$X \subset \mathbf{R}^n$, $Y \subset \mathbf{R}^m$;*
- *$f : X \to \mathbf{R}^m$ be a continuous mapping such that $f(x) \in Y$ for every $x \in X$;*
- *$g : Y \to \mathbf{R}^k$ be a continuous mapping.*

Then the composite mapping

$$h(x) = g(f(x)) : X \to \mathbf{R}^k$$

is continuous on X.

A.5.3 Basic properties of continuous functions on $\mathbf{R}^n$

The basic properties of continuous functions on $\mathbf{R}^n$ can be summarized as follows:

Theorem A.5.2 *Let X be a nonempty closed and bounded subset of $\mathbf{R}^n$.*
(i) *If a mapping $f : X \to \mathbf{R}^m$ is continuous on X, it is bounded on X: there exists $C < \infty$ such that $|f(x)| \le C$ for all $x \in X$.*

> Proof. Assume, on the contrary to what should be proved, that f is unbounded, so that for every i there exists a point $x^i \in X$ such that $|f(x^i)| > i$. By Theorem A.4.2, we can extract from the sequence $\{x^i\}$ a subsequence $\{x^{i_j}\}_{j=1}^{\infty}$ which converges to a point $\bar{x} \in X$. The real-valued function $g(x) = |f(x)|$ is continuous (as the superposition of two continuous mappings, see Theorem A.5.1.(ii)) and therefore its values at the points x^{i_j} should converge, as $j \to \infty$, to its value at $\bar{x}$; on the other hand, $g(x^{i_j}) \ge i_j \to \infty$ as $j \to \infty$, and we get the desired contradiction.

(ii) *If a mapping $f : X \to \mathbf{R}^m$ is continuous on X, it is uniformly continuous: for every $\epsilon > 0$ there exists $\delta > 0$ such that*

$$x, y \in X, |x - y| < \delta \Rightarrow |f(x) - f(y)| < \epsilon.$$

Proof. Assume, on the contrary to what should be proved, that there exists $\epsilon > 0$ such that for every $\delta > 0$ one can find a pair of points x, y in X such that $|x - y| < \delta$ and $|f(x) - f(y)| \geq \epsilon$. In particular, for every $i = 1, 2, ...$ we can find two points x^i, y^i in X such that $|x^i - y^i| \leq 1/i$ and $|f(x^i) - f(y^i)| \geq \epsilon$. By Theorem A.4.2, we can extract from the sequence $\{x^i\}$ a subsequence $\{x^{i_j}\}_{j=1}^{\infty}$ which converges to certain point $\bar{x} \in X$. Since $|y^{i_j} - x^{i_j}| \leq 1/i_j \to 0$ as $j \to \infty$, the sequence $\{y^{i_j}\}_{j=1}^{\infty}$ converges to the same point $\bar{x}$ as the sequence $\{x^{i_j}\}_{j=1}^{\infty}$ (why?) Since f is continuous, we have

$$\lim_{j\to\infty} f(y^{i_j}) = f(\bar{x}) = \lim_{j\to\infty} f(x^{i_j}),$$

whence $\lim\limits_{j\to\infty} (f(x^{i_j}) - f(y^{i_j})) = 0$, which contradicts the fact that $|f(x^{i_j}) - f(y^{i_j})| \geq \epsilon > 0$ for all j.

(iii) *Let f be a real-valued continuous function on X. The f attains its minimum on X:*

$$\operatorname*{Argmin}_X f \equiv \{x \in X : f(x) = \inf_{y\in X} f(y)\} \neq \emptyset,$$

same as f attains its maximum at certain points of X:

$$\operatorname*{Argmax}_X f \equiv \{x \in X : f(x) = \sup_{y\in X} f(y)\} \neq \emptyset.$$

Proof: Let us prove that f attains its maximum on X (the proof for minimum is completely similar). Since f is bounded on X by (i), the quantity

$$f^* = \sup_{x\in X} f(x)$$

is finite; of course, we can find a sequence $\{x^i\}$ of points from X such that $f^* = \lim\limits_{i\to\infty} f(x^i)$. By Theorem A.4.2, we can extract from the sequence $\{x^i\}$ a subsequence $\{x^{i_j}\}_{j=1}^{\infty}$ which converges to certain point $\bar{x} \in X$. Since f is continuous on X, we have

$$f(\bar{x}) = \lim_{j\to\infty} f(x^{i_j}) = \lim_{i\to\infty} f(x^i) = f^*,$$

so that the maximum of f on X indeed is achieved (e.g., at the point $\bar{x}$).

Exercise A.6 *Prove that in general no one of the three statements in Theorem A.5.2 remains valid when X is closed, but not bounded, same as when X is bounded, but not closed.*

A.6 Differentiable functions on $\mathbf{R}^n$

A.6.1 The derivative

The reader definitely is familiar with the notion of derivative of a real-valued function $f(x)$ of real variable x:

$$f'(x) = \lim_{\Delta x\to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

This definition does not work when we pass from functions of single real variable to functions of several real variables, or, which is the same, to functions with vector arguments. Indeed, in this case the shift in the argument Δx should be a vector, and we do not know what does it mean to *divide* by a vector...

A proper way to extend the notion of the derivative to real- and vector-valued functions of vector argument is to realize what in fact is the meaning of the derivative in the univariate case. What $f'(x)$ says to us is *how to approximate f in a neighbourhood of x by a linear function*. Specifically, *if* $f'(x)$

exists, then the linear function $f'(x)\Delta x$ of Δx approximates the change $f(x+\Delta x)-f(x)$ in f up to a remainder which is of highest order as compared with Δx as $\Delta x \to 0$:

$$|f(x+\Delta x)-f(x)-f'(x)\Delta x| \le \bar{o}(|\Delta x|) \text{ as } \Delta x \to 0.$$

In the above formula, we meet with the notation $\bar{o}(|\Delta x|)$, and here is the explanation of this notation:

> *$\bar{o}(|\Delta x|)$ is a common name of all functions $\phi(\Delta x)$ of Δx which are well-defined in a neighbourhood of the point $\Delta x = 0$ on the axis, vanish at the point $\Delta x = 0$ and are such that*
>
> $$\frac{\phi(\Delta x)}{|\Delta x|} \to 0 \text{ as } \Delta x \to 0.$$
>
> For example,
>
> 1. $(\Delta x)^2 = \bar{o}(|\Delta x|)$, $\Delta x \to 0$,
> 2. $|\Delta x|^{1.01} = \bar{o}(|\Delta x|)$, $\Delta x \to 0$,
> 3. $\sin^2(\Delta x) = \bar{o}(|\Delta x|)$, $\Delta x \to 0$,
> 4. $\Delta x \neq \bar{o}(|\Delta x|)$, $\Delta x \to 0$.
>
> Later on we shall meet with the notation "$\bar{o}(|\Delta x|^k)$ as $\Delta x \to 0$", where k is a positive integer. The definition is completely similar to the one for the case of $k = 1$:
>
> *$\bar{o}(|\Delta x|^k)$ is a common name of all functions $\phi(\Delta x)$ of Δx which are well-defined in a neighbourhood of the point $\Delta x = 0$ on the axis, vanish at the point $\Delta x = 0$ and are such that*
>
> $$\frac{\phi(\Delta x)}{|\Delta x|^k} \to 0 \text{ as } \Delta x \to 0.$$

Note that if $f(\cdot)$ is a function defined in a neighbourhood of a point x on the axis, then there perhaps are many linear functions $a\Delta x$ of Δx which well approximate $f(x+\Delta x)-f(x)$, in the sense that the remainder in the approximation

$$f(x+\Delta x)-f(x)-a\Delta x$$

tends to 0 as $\Delta x \to 0$; among these approximations, however, there exists *at most one* which approximates $f(x+\Delta x)-f(x)$ "very well" – so that the remainder is $\bar{o}(|\Delta x|)$, and not merely tends to 0 as $\Delta x \to 0$. Indeed, if

$$f(x+\Delta x)-f(x)-a\Delta x = \bar{o}(|\Delta x|),$$

then, dividing both sides by Δx, we get

$$\frac{f(x+\Delta x)-f(x)}{\Delta x} - a = \frac{\bar{o}(|\Delta x|)}{\Delta x};$$

by definition of $\bar{o}(\cdot)$, the right hand side in this equality tends to 0 as $\Delta x \to 0$, whence

$$a = \lim_{\Delta x \to 0} \frac{f(x+\Delta x)-f(x)}{\Delta x} = f'(x).$$

Thus, *if* a linear function $a\Delta x$ of Δx approximates the change $f(x+\Delta x)-f(x)$ in f up to the remainder which is $\bar{o}(|\Delta x|)$ as $\Delta x \to 0$, *then* a is the derivative of f at x. You can easily verify that the inverse statement also is true: *if* the derivative of f at x exists, *then* the linear function $f'(x)\Delta x$ of Δx approximates the change $f(x+\Delta x)-f(x)$ in f up to the remainder which is $\bar{o}(|\Delta x|)$ as $\Delta x \to 0$.

The advantage of the "$\bar{o}(|\Delta x|)$"-definition of derivative is that it can be naturally extended onto vector-valued functions of vector arguments (you should just replace "axis" with $\mathbf{R}^n$ in the definition of $\bar{o}$) and enlightens the *essence* of the notion of derivative: when it exists, this is exactly *the linear function of Δx which approximates the change $f(x+\Delta x)-f(x)$ in f up to a remainder which is $\bar{o}(|\Delta x|)$*. The precise definition is as follows:

Definition A.6.1 [Frechet differentiability] *Let f be a function which is well-defined in a neighbourhood of a point $x \in \mathbf{R}^n$ and takes values in $\mathbf{R}^m$. We say that f is differentiable at x, if there exists a linear function $Df(x)[\Delta x]$ of $\Delta x \in \mathbf{R}^n$ taking values in $\mathbf{R}^m$ which approximates the change $f(x+\Delta x)-f(x)$ in f up to a remainder which is $\bar{o}(|\Delta x|)$:*

$$|f(x+\Delta x)-f(x)-Df(x)[\Delta x]| \le \bar{o}(|\Delta x|). \tag{A.6.1}$$

Equivalently: a function f which is well-defined in a neighbourhood of a point $x \in \mathbf{R}^n$ and takes values in $\mathbf{R}^m$ is called differentiable at x, if there exists a linear function $Df(x)[\Delta x]$ of $\Delta x \in \mathbf{R}^n$ taking values in $\mathbf{R}^m$ such that for every $\epsilon > 0$ there exists $\delta > 0$ satisfying the relation

$$|\Delta x| \le \delta \Rightarrow |f(x+\Delta x)-f(x)-Df(x)[\Delta x]| \le \epsilon |\Delta x|.$$

A.6.2 Derivative and directional derivatives

We have defined what does it mean that a function $f : \mathbf{R}^n \to \mathbf{R}^m$ is differentiable at a point x, but did not say yet what is the derivative. The reader could guess that the derivative is exactly the "linear function $Df(x)[\Delta x]$ of $\Delta x \in \mathbf{R}^n$ taking values in $\mathbf{R}^m$ which approximates the change $f(x+\Delta x)-f(x)$ in f up to a remainder which is $\le \bar{o}(|\Delta x|)$" participating in the definition of differentiability. The guess is correct, but we cannot merely call the entity participating in the definition the derivative – why do we know that this entity is unique? Perhaps there are many different linear functions of Δx approximating the change in f up to a remainder which is $\bar{o}(|\Delta x|)$. In fact there is no more than a single linear function with this property due to the following observation:

> *Let f be differentiable at x, and $Df(x)[\Delta x]$ be a linear function participating in the definition of differentiability. Then*
>
> $$\forall \Delta x \in \mathbf{R}^n : \quad Df(x)[\Delta x] = \lim_{t \to +0} \frac{f(x+t\Delta x)-f(x)}{t}. \tag{A.6.2}$$
>
> *In particular, the derivative $Df(x)[\cdot]$ is uniquely defined by f and x.*
>
> **Proof.** We have
>
> $$\begin{array}{ll}
> |f(x+t\Delta x)-f(x)-Df(x)[t\Delta x]| \le \bar{o}(|t\Delta x|) & \\
> \Downarrow & \\
> |\frac{f(x+t\Delta x)-f(x)}{t} - \frac{Df(x)[t\Delta x]}{t}| \le \frac{\bar{o}(|t\Delta x|)}{t} & \\
> \Updownarrow & \\
> |\frac{f(x+t\Delta x)-f(x)}{t} - Df(x)[\Delta x]| \le \frac{\bar{o}(|t\Delta x|)}{t} & [\text{since } Df(x)[\cdot] \text{ is linear}] \\
> \Downarrow & \\
> Df(x)[\Delta x] = \lim_{t \to +0} \frac{f(x+t\Delta x)-f(x)}{t} & \left[\begin{array}{l}\text{passing to limit as } t \to +0; \\ \text{note that } \frac{\bar{o}(|t\Delta x|)}{t} \to 0, t \to +0\end{array}\right]
> \end{array}$$

Pay attention to important remarks as follows:

1. The right hand side limit in (A.6.2) is an important entity called the *directional derivative of f taken at x along (a direction) Δx*; note that this quantity is defined in the "purely univariate" fashion – by dividing the change in f by the magnitude of a shift in a direction Δx and passing to limit as the magnitude of the shift approaches 0. Relation (A.6.2) says that the derivative, if exists, is, at every Δx, nothing that the directional derivative of f taken at x along Δx. Note, however, that differentiability is much more than the existence of directional derivatives along all directions Δx; differentiability requires also *the directional derivatives to be "well-organized" – to depend linearly on the direction Δx.* It is easily seen that just existence of directional derivatives does not imply their "good organization": for example, the Euclidean norm

$$f(x) = |x|$$

at $x = 0$ possesses directional derivatives along all directions:

$$\lim_{t\to+0} \frac{f(0+t\Delta x) - f(0)}{t} = |\Delta x|;$$

these derivatives, however, depend *non-linearly* on Δx, so that the Euclidean norm is *not* differentiable at the origin (although is differentiable everywhere outside the origin, but this is another story).

2. It should be stressed that the derivative, if exists, is what it is: *a linear function of $\Delta x \in \mathbf{R}^n$ taking values in $\mathbf{R}^m$*. As we shall see in a while, we can *represent* this function by something "tractable", like a vector or a matrix, and can understand how to compute such a representation; however, an intelligent reader should bear in mind that *a* representation is not exactly the same as *the* represented entity. Sometimes the difference between derivatives and the entities which represent them is reflected in the terminology: what we call the *derivative*, is also called the *differential*, while the word "derivative" is reserved for the vector/matrix representing the differential.

A.6.3 Representations of the derivative

indexderivatives!representation ofBy definition, the derivative of a mapping $f : \mathbf{R}^n \to \mathbf{R}^m$ at a point x is a linear function $Df(x)[\Delta x]$ taking values in $\mathbf{R}^m$. How could we represent such a function?

Case of $m = 1$ – the gradient. Let us start with real-valued functions (i.e., with the case of $m = 1$); in this case the derivative is a *linear* real-valued function on $\mathbf{R}^n$. As we remember, the standard Euclidean structure on $\mathbf{R}^n$ allows to represent every linear function on $\mathbf{R}^n$ as the inner product of the argument with certain fixed vector. In particular, the derivative $Df(x)[\Delta x]$ of a scalar function can be represented as

$$Df(x)[\Delta x] = [\text{vector}]^T \Delta x;$$

what is denoted "vector" in this relation, is called the *gradient* of f at x and is denoted by $\nabla f(x)$:

$$Df(x)[\Delta x] = (\nabla f(x))^T \Delta x. \tag{A.6.3}$$

How to compute the gradient? The answer is given by (A.6.2). Indeed, let us look what (A.6.3) and (A.6.2) say when Δx is the i-th standard basic orth. According to (A.6.3), $Df(x)[e_i]$ is the i-th coordinate of the vector $\nabla f(x)$; according to (A.6.2),

$$\left.\begin{array}{rl} Df(x)[e_i] &= \lim\limits_{t\to+0} \frac{f(x+te_i)-f(x)}{t}, \\ Df(x)[e_i] = -Df(x)[-e_i] &= -\lim\limits_{t\to+0} \frac{f(x-te_i)-f(x)}{t} = \lim\limits_{t\to-0} \frac{f(x+te_i)-f(x)}{t} \end{array}\right\} \Rightarrow Df(x)[e_i] = \frac{\partial f(x)}{\partial x_i}.$$

Thus,

If a real-valued function f is differentiable at x, then the first order partial derivatives of f at x exist, and the gradient of f at x is just the vector with the coordinates which are the first order partial derivatives of f taken at x:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}.$$

The derivative of f, taken at x, is the linear function of Δx given by

$$Df(x)[\Delta x] = (\nabla f(x))^T \Delta x = \sum_{i=1}^{n} \frac{\partial f(x)}{\partial x_i} (\Delta x)_i.$$

General case – the Jacobian. Now let $f : \mathbf{R}^n \to \mathbf{R}^m$ with $m \geq 1$. In this case, $Df(x)[\Delta x]$, regarded as a function of Δx, is a linear mapping from $\mathbf{R}^n$ to $\mathbf{R}^m$; as we remember, the standard way to represent a linear mapping from $\mathbf{R}^n$ to $\mathbf{R}^m$ is to represent it as the multiplication by $m \times n$ matrix:

$$Df(x)[\Delta x] = [m \times n \text{ matrix}] \cdot \Delta x. \tag{A.6.4}$$

What is denoted by "matrix" in (A.6.4), is called *the Jacobian* of f at x and is denoted by $f'(x)$. How to compute the entries of the Jacobian? Here again the answer is readily given by (A.6.2). Indeed, on one hand, we have

$$Df(x)[\Delta x] = f'(x)\Delta x, \tag{A.6.5}$$

whence

$$[Df(x)[e_j]]_i = (f'(x))_{ij},\ i = 1, ..., m, j = 1, ..., n.$$

On the other hand, denoting

$$f(x) = \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix},$$

the same computation as in the case of gradient demonstrates that

$$[Df(x)[e_j]]_i = \frac{\partial f_i(x)}{\partial x_j}$$

and we arrive at the following conclusion:

> *If a vector-valued function $f(x) = (f_1(x), ..., f_m(x))$ is differentiable at x, then the first order partial derivatives of all f_i at x exist, and the Jacobian of f at x is just the $m \times n$ matrix with the entries $[\frac{\partial f_i(x)}{\partial x_j}]_{i,j}$ (so that the rows in the Jacobian are $[\nabla f_1(x)]^T$,..., $[\nabla f_m(x)]^T$. The derivative of f, taken at x, is the linear vector-valued function of Δx given by*
>
> $$Df(x)[\Delta x] = f'(x)\Delta x = \begin{bmatrix} [\nabla f_1(x)]^T \Delta x \\ \vdots \\ [\nabla f_m(x)]^T \Delta x \end{bmatrix}.$$

Remark A.6.1 Note that for a real-valued function f we have defined both the gradient $\nabla f(x)$ and the Jacobian $f'(x)$. These two entities are "nearly the same", but not exactly the same: the Jacobian is a vector-row, and the gradient is a vector-column linked by the relation

$$f'(x) = (\nabla f(x))^T.$$

Of course, both these representations of the derivative of f yield the same linear approximation of the change in f:

$$Df(x)[\Delta x] = (\nabla f(x))^T \Delta x = f'(x)\Delta x.$$

A.6.4 Existence of the derivative

We have seen that the existence of the derivative of f at a point implies the existence of the first order partial derivatives of the (components $f_1, ..., f_m$ of) f. The inverse statement is not exactly true – the existence of all first order partial derivatives $\frac{\partial f_i(x)}{\partial x_j}$ not necessarily implies the existence of the derivative; we need a bit more:

Theorem A.6.1 [Sufficient condition for differentiability] *Assume that*

1. *The mapping $f = (f_1, ..., f_m) : \mathbf{R}^n \to \mathbf{R}^m$ is well-defined in a neighbourhood U of a point $x_0 \in \mathbf{R}^n$,*
2. *The first order partial derivatives of the components f_i of f exist everywhere in U, and*
3. *The first order partial derivatives of the components f_i of f are continuous at the point x_0.*

Then f is differentiable at the point x_0.

A.6.5 Calculus of derivatives

The calculus of derivatives is given by the following result:

Theorem A.6.2 (i) [Differentiability and linear operations] *Let $f_1(x)$, $f_2(x)$ be mappings defined in a neighbourhood of $x_0 \in \mathbf{R}^n$ and taking values in $\mathbf{R}^m$, and $\lambda_1(x), \lambda_2(x)$ be real-valued functions defined in a neighbourhood of x_0. Assume that $f_1, f_2, \lambda_1, \lambda_2$ are differentiable at x_0. Then so is the function $f(x) = \lambda_1(x)f_1(x) + \lambda_2(x)f_2(x)$, with the derivative at x_0 given by*

$$\begin{array}{rcl} Df(x_0)[\Delta x] & = & [D\lambda_1(x_0)[\Delta x]]f_1(x_0) + \lambda_1(x_0)Df_1(x_0)[\Delta x] \\ & & +[D\lambda_2(x_0)[\Delta x]]f_2(x_0) + \lambda_2(x_0)Df_2(x_0)[\Delta x] \\ & & \Downarrow \\ f'(x_0) & = & f_1(x_0)[\nabla\lambda_1(x_0)]^T + \lambda_1(x_0)f_1'(x_0) \\ & & +f_2(x_0)[\nabla\lambda_2(x_0)]^T + \lambda_2(x_0)f_2'(x_0). \end{array}$$

(ii) [chain rule] *Let a mapping $f : \mathbf{R}^n \to \mathbf{R}^m$ be differentiable at x_0, and a mapping $g : \mathbf{R}^m \to \mathbf{R}^n$ be differentiable at $y_0 = f(x_0)$. Then the superposition $h(x) = g(f(x))$ is differentiable at x_0, with the derivative at x_0 given by*

$$\begin{array}{c} Dh(x_0)[\Delta x] = Dg(y_0)[Df(x_0)[\Delta x]] \\ \Downarrow \\ h'(x_0) = g'(y_0)f'(x_0) \end{array}$$

If the outer function g is real-valued, then the latter formula implies that

$$\nabla h(x_0) = [f'(x_0)]^T \nabla g(y_0)$$

(recall that for a real-valued function ϕ, $\phi' = (\nabla\phi)^T$).

A.6.6 Computing the derivative

Representations of the derivative via first order partial derivatives normally allow to compute it by the standard Calculus rules, in a completely mechanical fashion, not thinking at all of *what* we are computing. The examples to follow (especially the third of them) demonstrate that it often makes sense to bear in mind *what* is the derivative; this sometimes yield the result much faster than blind implementing Calculus rules.

Example 1: The gradient of an affine function. An *affine* function

$$f(x) = a + \sum_{i=1}^n g_i x_i \equiv a + g^T x : \mathbf{R}^n \to \mathbf{R}$$

is differentiable at every point (Theorem A.6.1) and its gradient, of course, equals g:

$$\begin{array}{rcll} (\nabla f(x))^T \Delta x & = & \lim\limits_{t \to +0} t^{-1} \left[f(x + t\Delta x) - f(x)\right] & [(A.6.2)] \\ & = & \lim\limits_{t \to +0} t^{-1}[tg^T \Delta x] & [\text{arithmetics}] \end{array}$$

and we arrive at

$$\boxed{\nabla(a + g^T x) = g}$$

Example 2: The gradient of a quadratic form. For the time being, let us define a homogeneous quadratic form on $\mathbf{R}^n$ as a function

$$f(x) = \sum_{i,j} A_{ij} x_i x_j = x^T A x,$$

where A is an $n \times n$ matrix. Note that the matrices A and A^T define the same quadratic form, and therefore the *symmetric* matrix $B = \frac{1}{2}(A + A^T)$ also produces the same quadratic form as A and A^T. It follows that we always may assume (and do assume from now on) that the matrix A producing the quadratic form in question is symmetric.

A quadratic form is a simple polynomial and as such is differentiable at every point (Theorem A.6.1). What is the gradient of f at a point x? Here is the computation:

$$\begin{array}{rcll}
(\nabla f(x))^T \Delta x & = & Df(x)[\Delta x] & \\
& = & \lim\limits_{t \to +0} \left[(x + t\Delta x)^T A(x + t\Delta x) - x^T A x\right] & \\
& & & [(A.6.2)] \\
& = & \lim\limits_{t \to +0} \left[x^T A x + t(\Delta x)^T A x + t x^T A \Delta x + t^2 (\Delta x)^T A \Delta x - x^T A x\right] & \\
& & & [\text{opening parentheses}] \\
& = & \lim\limits_{t \to +0} t^{-1} \left[2t(Ax)^T \Delta x + t^2 (\Delta x)^T A \Delta x\right] & \\
& & & [\text{arithmetics + symmetry of } A] \\
& = & 2(Ax)^T \Delta x &
\end{array}$$

We conclude that

$$\boxed{\nabla(x^T A x) = 2Ax}$$

(recall that $A = A^T$).

Example 3: The derivative of the log-det barrier. Let us compute the derivative of the *log-det barrier* (playing an extremely important role in modern optimization)

$$F(X) = \ln \mathrm{Det}\,(X);$$

here X is an $n \times n$ matrix (or, if you prefer, n^2-dimensional vector). Note that $F(X)$ is well-defined and differentiable in a neighbourhood of every point $\bar{X}$ with positive determinant (indeed, $\mathrm{Det}\,(X)$ is a polynomial of the entries of X and thus – is everywhere continuous and differentiable with continuous partial derivatives, while the function $\ln(t)$ is continuous and differentiable on the positive ray; by Theorems A.5.1.(ii), A.6.2.(ii), F is differentiable at every X such that $\mathrm{Det}\,(X) > 0$). The reader is kindly asked to try to find the derivative of F by the standard techniques; if the result will not be obtained in, say, 30 minutes, please look at the 8-line computation to follow (in this computation, $\mathrm{Det}\,(\bar{X}) > 0$, and $G(X) = \mathrm{Det}\,(X)$):

$$\begin{array}{rcll}
 & DF(\bar{X})[\Delta X] & & \\
= & D\ln(G(\bar{X}))[DG(\bar{X})[\Delta X]] & & [\text{chain rule}] \\
= & G^{-1}(\bar{X}) DG(\bar{X})[\Delta X] & & [\ln'(t) = t^{-1}] \\
= & \mathrm{Det}^{-1}(\bar{X}) \lim\limits_{t \to +0} t^{-1} \left[\mathrm{Det}\,(\bar{X} + t\Delta X) - \mathrm{Det}\,(\bar{X})\right] & & [\text{definition of } G \text{ and (A.6.2)}] \\
= & \mathrm{Det}^{-1}(\bar{X}) \lim\limits_{t \to +0} t^{-1} \left[\mathrm{Det}\,(\bar{X}(I + t\bar{X}^{-1}\Delta X)) - \mathrm{Det}\,(\bar{X})\right] & & \\
= & \mathrm{Det}^{-1}(\bar{X}) \lim\limits_{t \to +0} t^{-1} \left[\mathrm{Det}\,(\bar{X})(\mathrm{Det}\,(I + t\bar{X}^{-1}\Delta X) - 1)\right] & & [\mathrm{Det}\,(AB) = \mathrm{Det}\,(A)\,\mathrm{Det}\,(B)] \\
= & \lim\limits_{t \to +0} t^{-1} \left[\mathrm{Det}\,(I + t\bar{X}^{-1}\Delta X) - 1\right] & & \\
= & \mathrm{Tr}(\bar{X}^{-1}\Delta X) = \sum\limits_{i,j} [\bar{X}^{-1}]_{ji} (\Delta X)_{ij} & &
\end{array}$$

where the concluding equality

$$\lim_{t \to +0} t^{-1} [\mathrm{Det}\,(I + tA) - 1] = \mathrm{Tr}(A) \equiv \sum_i A_{ii} \tag{A.6.6}$$

is immediately given by recalling what is the determinant of $I + tA$: this is a polynomial of t which is the sum of products, taken along all diagonals of a $n \times n$ matrix and assigned certain signs, of the entries of $I + tA$. At every one of these diagonals, except for the main one, there are at least two cells with the entries

proportional to t, so that the corresponding products do not contribute to the constant and the linear in t terms in $\mathrm{Det}\,(I+tA)$ and thus do not affect the limit in (A.6.6). The only product which does contribute to the linear and the constant terms in $\mathrm{Det}\,(I+tA)$ is the product $(1+tA_{11})(1+tA_{22})...(1+tA_{nn})$ coming from the main diagonal; it is clear that in this product the constant term is 1, and the linear in t term is $t(A_{11}+...+A_{nn})$, and (A.6.6) follows.

A.6.7 Higher order derivatives

Let $f:\mathbf{R}^n\to\mathbf{R}^m$ be a mapping which is well-defined and differentiable at every point x from an open set U. The Jacobian of this mapping $J(x)$ is a mapping from $\mathbf{R}^n$ to the space $\mathbf{R}^{m\times n}$ matrices, i.e., is a mapping taking values in certain $\mathbf{R}^M$ ($M=mn$). The derivative of this mapping, if it exists, is called the *second derivative* of f; it again is a mapping from $\mathbf{R}^n$ to certain $\mathbf{R}^M$ and as such can be differentiable, and so on, so that we can speak about the second, the third, ... derivatives of a vector-valued function of vector argument. A *sufficient* condition for the existence of k derivatives of f in U is that f is C^k in U, i.e., that all partial derivatives of f of orders $\le k$ exist and are continuous everywhere in U (cf. Theorem A.6.1).

We have explained what does it mean that f has k derivatives in U; note, however, that according to the definition, highest order derivatives at a point x are just long vectors; say, the second order derivative of a scalar function f of 2 variables is the Jacobian of the mapping $x\mapsto f'(x):\mathbf{R}^2\to\mathbf{R}^2$, i.e., a mapping from $\mathbf{R}^2$ to $\mathbf{R}^{2\times 2}=\mathbf{R}^4$; the third order derivative of f is therefore the Jacobian of a mapping from $\mathbf{R}^2$ to $\mathbf{R}^4$, i.e., a mapping from $\mathbf{R}^2$ to $\mathbf{R}^{4\times 2}=\mathbf{R}^8$, and so on. The question which should be addressed now is: *What is a natural and transparent way to represent the highest order derivatives?*

The answer is as follows:

> (*) *Let $f:\mathbf{R}^n\to\mathbf{R}^m$ be C^k on an open set $U\subset\mathbf{R}^n$. The derivative of order $\ell\le k$ of f, taken at a point $x\in U$, can be naturally identified with a function*
>
> $$D^\ell f(x)[\Delta x^1,\Delta x^2,...,\Delta x^\ell]$$
>
> *of ℓ vector arguments $\Delta x^i\in\mathbf{R}^n$, $i=1,...,\ell$, and taking values in $\mathbf{R}^m$. This function is linear in every one of the arguments Δx^i, the other arguments being fixed, and is symmetric with respect to permutation of arguments $\Delta x^1,...,\Delta x^\ell$.*
>
> *In terms of f, the quantity $D^\ell f(x)[\Delta x^1,\Delta x^2,...,\Delta x^\ell]$ (full name: "the ℓ-th derivative (or differential) of f taken at a point x along the directions $\Delta x^1,...,\Delta x^\ell$") is given by*
>
> $$D^\ell f(x)[\Delta x^1,\Delta x^2,...,\Delta x^\ell]=\frac{\partial^\ell}{\partial t_\ell\partial t_{\ell-1}...\partial t_1}\Big|_{t_1=...=t_\ell=0}f(x+t_1\Delta x^1+t_2\Delta x^2+...+t_\ell\Delta x^\ell). \tag{A.6.7}$$

The explanation to our claims is as follows. Let $f:\mathbf{R}^n\to\mathbf{R}^m$ be C^k on an open set $U\subset\mathbf{R}^n$.

1. When $\ell=1$, (*) says to us that the first order derivative of f, taken at x, is a linear function $Df(x)[\Delta x^1]$ of $\Delta x^1\in\mathbf{R}^n$, taking values in $\mathbf{R}^m$, and that the value of this function at every Δx^1 is given by the relation
$$Df(x)[\Delta x^1]=\frac{\partial}{\partial t_1}\Big|_{t_1=0}f(x+t_1\Delta x^1) \tag{A.6.8}$$
(cf. (A.6.2)), which is in complete accordance with what we already know about the derivative.

2. To understand what is the second derivative, let us take the first derivative $Df(x)[\Delta x^1]$, *let us temporarily fix somehow the argument* Δx^1 and treat the derivative as a function of x. As a function of x, Δx^1 being fixed, the quantity $Df(x)[\Delta x^1]$ is again a mapping which maps U into $\mathbf{R}^m$ and is differentiable by Theorem A.6.1 (provided, of course, that $k\ge 2$). The derivative of this mapping is certain linear function of $\Delta x\equiv\Delta x^2\in\mathbf{R}^n$, depending on x as on a parameter; and of course it depends on Δx^1 as on a parameter as well. Thus, the derivative of $Df(x)[\Delta x^1]$ in x is certain function
$$D^2f(x)[\Delta x^1,\Delta x^2]$$

of $x \in U$ and $\Delta x^1, \Delta x^2 \in \mathbf{R}^n$ and taking values in $\mathbf{R}^m$. What we know about this function is that it is linear in Δx^2. In fact, it is also linear in Δx^1, since it is the derivative in x of certain function (namely, of $Df(x)[\Delta x^1]$) *linearly depending on the parameter* Δx^1, so that the derivative of the function *in* x is linear in the parameter Δx^1 as well (differentiation is a linear operation with respect to a function we are differentiating: summing up functions and multiplying them by real constants, we sum up, respectively, multiply by the same constants, the derivatives). Thus, $D^2f(x)[\Delta x^1, \Delta x^2]$ is linear in Δx^1 when x and Δx^2 are fixed, and is linear in Δx^2 when x and Δx^1 are fixed. Moreover, we have

$$\begin{array}{rcll} D^2f(x)[\Delta x^1, \Delta x^2] & = & \frac{\partial}{\partial t_2}\Big|_{t_2=0} Df(x+t_2\Delta x^2)[\Delta x^1] & \text{[cf. (A.6.8)]} \\ & = & \frac{\partial}{\partial t_2}\Big|_{t_2=0} \frac{\partial}{\partial t_1}\Big|_{t_1=0} f(x+t_2\Delta x^2+t_1\Delta x^1) & \text{[by (A.6.8)]} \\ & = & \frac{\partial^2}{\partial t_2 \partial t_1}\Big|_{t_1=t_2=0} f(x+t_1\Delta x^1+t_2\Delta x^2) & \end{array} \tag{A.6.9}$$

as claimed in (A.6.7) for $\ell = 2$. The only piece of information about the second derivative which is contained in $(*)$ and is not justified yet is that $D^2f(x)[\Delta x^1, \Delta x^2]$ is symmetric in $\Delta x^1, \Delta x^2$; but this fact is readily given by the representation (A.6.7), since, as they prove in Calculus, if a function ϕ possesses *continuous* partial derivatives of orders $\le \ell$ in a neighbourhood of a point, then these derivatives in this neighbourhood are independent of the order in which they are taken; it follows that

$$\begin{array}{rcll} D^2f(x)[\Delta x^1, \Delta x^2] & = & \frac{\partial^2}{\partial t_2 \partial t_1}\Big|_{t_1=t_2=0} \underbrace{f(x+t_1\Delta x^1+t_2\Delta x^2)}_{\phi(t_1,t_2)} & \text{[(A.6.9)]} \\ & = & \frac{\partial^2}{\partial t_1 \partial t_2}\Big|_{t_1=t_2=0} \phi(t_1,t_2) & \\ & = & \frac{\partial^2}{\partial t_1 \partial t_2}\Big|_{t_1=t_2=0} f(x+t_2\Delta x^2+t_1\Delta x^1) & \\ & = & D^2f(x)[\Delta x^2, \Delta x^1] & \text{[the same (A.6.9)]} \end{array}$$

3. Now it is clear how to proceed: to define $D^3f(x)[\Delta x^1, \Delta x^2, \Delta x^3]$, we fix in the second order derivative $D^2f(x)[\Delta x^1, \Delta x^2]$ the arguments $\Delta x^1, \Delta x^2$ and treat it as a function of x only, thus arriving at a mapping which maps U into $\mathbf{R}^m$ and depends on $\Delta x^1, \Delta x^2$ as on parameters (linearly in every one of them). Differentiating the resulting mapping in x, we arrive at a function $D^3f(x)[\Delta x^1, \Delta x^2, \Delta x^3]$ which by construction is linear in every one of the arguments Δx^1, Δx^2, Δx^3 and satisfies (A.6.7); the latter relation, due to the Calculus result on the symmetry of partial derivatives, implies that $D^3f(x)[\Delta x^1, \Delta x^2, \Delta x^3]$ is symmetric in $\Delta x^1, \Delta x^2, \Delta x^3$. After we have in our disposal the third derivative D^3f, we can build from it in the already explained fashion the fourth derivative, and so on, until k-th derivative is defined.

Remark A.6.2 Since $D^\ell f(x)[\Delta x^1, ..., \Delta x^\ell]$ is linear in every one of Δx^i, we can expand the derivative in a multiple sum:

$$\begin{array}{rcl} & & \Delta x^i = \sum_{j=1}^n \Delta x^i_j e_j \\ & & \Downarrow \\ D^\ell f(x)[\Delta x^1, ..., \Delta x^\ell] & = & D^\ell f(x)[\sum_{j_1=1}^n \Delta x^1_{j_1} e_{j_1}, ..., \sum_{j_\ell=1}^n \Delta x^\ell_{j_\ell} e_{j_\ell}] \\ & = & \sum_{1 \le j_1, ..., j_\ell \le n} D^\ell f(x)[e_{j_1}, ..., e_{j_\ell}] \Delta x^1_{j_1} ... \Delta x^\ell_{j_\ell} \end{array} \tag{A.6.10}$$

What is the origin of the coefficients $D^\ell f(x)[e_{j_1}, ..., e_{j_\ell}]$? According to (A.6.7), one has

$$\begin{array}{rcl} D^\ell f(x)[e_{j_1}, ..., e_{j_\ell}] & = & \frac{\partial^\ell}{\partial t_\ell \partial t_{\ell-1} ... \partial t_1}\Big|_{t_1=...=t_\ell=0} f(x+t_1e_{j_1}+t_2e_{j_2}+...+t_\ell e_{j_\ell}) \\ & = & \frac{\partial^\ell}{\partial x_{j_\ell} \partial x_{j_{\ell-1}} ... \partial x_{j_1}} f(x). \end{array}$$

so that the coefficients in (A.6.10) are nothing but the partial derivatives, of order ℓ, of f.

Remark A.6.3 An important particular case of relation (A.6.7) is the one when $\Delta x^1 = \Delta x^2 = ... = \Delta x^\ell$; let us call the common value of these ℓ vectors d. According to (A.6.7), we have

$$D^\ell f(x)[d,d,...,d] = \left.\frac{\partial^\ell}{\partial t_\ell \partial t_{\ell-1}...\partial t_1}\right|_{t_1=...=t_\ell=0} f(x+t_1 d + t_2 d + ... + t_\ell d).$$

This relation can be interpreted as follows: consider the function

$$\phi(t) = f(x+td)$$

of a real variable t. Then (check it!)

$$\phi^{(\ell)}(0) = \left.\frac{\partial^\ell}{\partial t_\ell \partial t_{\ell-1}...\partial t_1}\right|_{t_1=...=t_\ell=0} f(x+t_1 d + t_2 d + ... + t_\ell d) = D^\ell f(x)[d,...,d].$$

In other words, $D^\ell f(x)[d,...,d]$ is what is called *ℓ-th directional derivative of f taken at x along the direction d*; to define this quantity, we pass from function f of several variables to the univariate function $\phi(t) = f(x+td)$ – restrict f onto the line passing through x and directed by d – and then take the "usual" derivative of order ℓ of the resulting function of single real variable t at the point $t = 0$ (which corresponds to the point x of our line).

Representation of higher order derivatives. k-th order derivative $D^k f(x)[\cdot,...,\cdot]$ of a C^k function $f : \mathbf{R}^n \to \mathbf{R}^m$ is what it is – it is a symmetric k-linear mapping on $\mathbf{R}^n$ taking values in $\mathbf{R}^m$ and depending on x as on a parameter. Choosing somehow coordinates in $\mathbf{R}^n$, we can represent such a mapping in the form

$$D^k f(x)[\Delta x_1,...,\Delta x_k] = \sum_{1\le i_1,...,i_k\le n} \frac{\partial^k f(x)}{\partial x_{i_k}\partial x_{i_{k-1}}...\partial x_{i_1}} (\Delta x_1)_{i_1}...(\Delta x_k)_{i_k}.$$

We may say that the derivative can be represented by k-index collection of m-dimensional vectors $\frac{\partial^k f(x)}{\partial x_{i_k}\partial x_{i_{k-1}}...\partial x_{i_1}}$. This collection, however, is a difficult-to-handle entity, so that such a representation does not help. There is, however, a case when the collection becomes an entity we know to handle; this is the case of the second-order derivative of a scalar function ($k = 2, m = 1$). In this case, the collection in question is just a symmetric matrix $H(x) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j}\right]_{1\le i,j\le n}$. This matrix is called the *Hessian* of f at x. Note that

$$D^2 f(x)[\Delta x_1, \Delta x_2] = \Delta x_1^T H(x) \Delta x_2.$$

A.6.8 Calculus of C^k mappings

The calculus of C^k mappings can be summarized as follows:

Theorem A.6.3 (i) *Let U be an open set in $\mathbf{R}^n$, $f_1(\cdot), f_2(\cdot) : \mathbf{R}^n \to \mathbf{R}^m$ be C^k in U, and let real-valued functions $\lambda_1(\cdot), \lambda_2(\cdot)$ be C^k in U. Then the function*

$$f(x) = \lambda_1(x) f_1(x) + \lambda_2(x) f_2(x)$$

is C^k in U.

(ii) *Let U be an open set in $\mathbf{R}^n$, V be an open set in $\mathbf{R}^m$, let a mapping $f : \mathbf{R}^n \to \mathbf{R}^m$ be C^k in U and such that $f(x) \in V$ for $x \in U$, and, finally, let a mapping $g : \mathbf{R}^m \to \mathbf{R}^p$ be C^k in V. Then the superposition*

$$h(x) = g(f(x))$$

is C^k in U.

Remark A.6.4 For higher order derivatives, in contrast to the first order ones, there is no simple "chain rule" for computing the derivative of superposition. For example, the second-order derivative of the superposition $h(x) = g(f(x))$ of two C^2-mappings is given by the formula

$$Dh(x)[\Delta x^1, \Delta x^2] = Dg(f(x))[D^2 f(x)[\Delta x^1, \Delta x^2]] + D^2 g(x)[Df(x)[\Delta x^1], Df(x)[\Delta x^2]]$$

(check it!). We see that both the first- and the second-order derivatives of f and g contribute to the second-order derivative of the superposition h.

The only case when there does exist a simple formula for high order derivatives of a superposition is the case when the inner function is affine: if $f(x) = Ax + b$ and $h(x) = g(f(x)) = g(Ax + b)$ with a C^ℓ mapping g, then

$$D^\ell h(x)[\Delta x^1, ..., \Delta x^\ell] = D^\ell g(Ax + b)[A\Delta x^1, ..., A\Delta x^\ell]. \tag{A.6.11}$$

A.6.9 Examples of higher-order derivatives

Example 1: Second-order derivative of an affine function $f(x) = a + b^T x$ is, of course, identically zero. Indeed, as we have seen,

$$Df(x)[\Delta x^1] = b^T \Delta x^1$$

is independent of x, and therefore the derivative of $Df(x)[\Delta x^1]$ in x, which should give us the second derivative $D^2 f(x)[\Delta x^1, \Delta x^2]$, is zero. Clearly, the third, the fourth, etc., derivatives of an affine function are zero as well.

Example 2: Second-order derivative of a homogeneous quadratic form $f(x) = x^T Ax$ (A is a symmetric $n \times n$ matrix). As we have seen,

$$Df(x)[\Delta x^1] = 2x^T A\Delta x^1.$$

Differentiating in x, we get

$$D^2 f(x)[\Delta x^1, \Delta x^2] = \lim_{t \to +0} t^{-1} \left[2(x + t\Delta x^2)^T A\Delta x^1 - 2x^T A\Delta x^1\right] = 2(\Delta x^2)^T A\Delta x^1,$$

so that

$$\boxed{D^2 f(x)[\Delta x^1, \Delta x^2] = 2(\Delta x^2)^T A\Delta x^1}$$

Note that the second derivative of a quadratic form is independent of x; consequently, the third, the fourth, etc., derivatives of a quadratic form are identically zero.

Example 3: Second-order derivative of the log-det barrier $F(X) = \ln \mathrm{Det}\,(X)$. As we have seen, this function of an $n \times n$ matrix is well-defined and differentiable on the set U of matrices with positive determinant (which is an open set in the space $\mathbf{R}^{n \times n}$ of $n \times n$ matrices). In fact, this function is C^∞ in U. Let us compute its second-order derivative. As we remember,

$$DF(X)[\Delta X^1] = \mathrm{Tr}(X^{-1}\Delta X^1). \tag{A.6.12}$$

To differentiate the right hand side in X, let us first find the derivative of the mapping $G(X) = X^{-1}$ which is defined on the open set of non-degenerate $n \times n$ matrices. We have

$$\begin{array}{rcl} DG(X)[\Delta X] & = & \lim\limits_{t\to+0} t^{-1}\left[(X+t\Delta X)^{-1} - X^{-1}\right] \\ & = & \lim\limits_{t\to+0} t^{-1}\left[(X(I+tX^{-1}\Delta X))^{-1} - X^{-1}\right] \\ & = & \lim\limits_{t\to+0} t^{-1}\big[(I+t\underbrace{X^{-1}\Delta X}_{Y})^{-1}X^{-1} - X^{-1}\big] \\ & = & \left[\lim\limits_{t\to+0} t^{-1}\left[(I+tY)^{-1} - I\right]\right] X^{-1} \\ & = & \left[\lim\limits_{t\to+0} t^{-1}\left[I-(I+tY)\right](I+tY)^{-1}\right] X^{-1} \\ & = & \left[\lim\limits_{t\to+0}\left[-Y(I+tY)^{-1}\right]\right] X^{-1} \\ & = & -YX^{-1} \\ & = & -X^{-1}\Delta X X^{-1} \end{array}$$

and we arrive at the important by its own right relation

$$\boxed{D(X^{-1})[\Delta X] = -X^{-1}\Delta X X^{-1}, \quad [X \in \mathbf{R}^{n\times n}, \mathrm{Det}\,(X) \neq 0]}$$

which is the "matrix extension" of the standard relation $(x^{-1})' = -x^{-2}$, $x \in \mathbf{R}$.

Now we are ready to compute the second derivative of the log-det barrier:

$$\begin{array}{rcl} & & F(X) = \ln \mathrm{Det}\,(X) \\ & & \Downarrow \\ & & DF(X)[\Delta X^1] = \mathrm{Tr}(X^{-1}\Delta X^1) \\ & & \Downarrow \\ D^2F(X)[\Delta X^1, \Delta X^2] & = & \lim\limits_{t\to+0} t^{-1}\left[\mathrm{Tr}((X+t\Delta X^2)^{-1}\Delta X^1) - \mathrm{Tr}(X^{-1}\Delta X^1)\right] \\ & = & \lim\limits_{t\to+0} \mathrm{Tr}\left(t^{-1}(X+t\Delta X^2)^{-1}\Delta X^1 - X^{-1}\Delta X^1\right) \\ & = & \lim\limits_{t\to+0} \mathrm{Tr}\left(\left[t^{-1}(X+t\Delta X^2)^{-1} - X^{-1}\right]\Delta X^1\right) \\ & = & \mathrm{Tr}\left([-X^{-1}\Delta X^2 X^{-1}]\Delta X^1\right), \end{array}$$

and we arrive at the formula

$$\boxed{D^2F(X)[\Delta X^1, \Delta X^2] = -\mathrm{Tr}(X^{-1}\Delta X^2 X^{-1}\Delta X^1) \quad [X \in \mathbf{R}^{n\times n}, \mathrm{Det}\,(X) > 0]}$$

Since $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ (check it!) for all matrices A, B such that the product AB makes sense and is square, the right hand side in the above formula is symmetric in ΔX^1, ΔX^2, as it should be for the second derivative of a C^2 function.

A.6.10 Taylor expansion

Assume that $f : \mathbf{R}^n \to \mathbf{R}^m$ is C^k in a neighbourhood U of a point $\bar{x}$. The *Taylor expansion of order k* of f, built at the point $\bar{x}$, is the function

$$\begin{array}{rcl} F_k(x) & = & f(\bar{x}) + \frac{1}{1!}Df(\bar{x})[x-\bar{x}] + \frac{1}{2!}D^2f(\bar{x})[x-\bar{x}, x-\bar{x}] \\ & & + \frac{1}{3!}D^2f(\bar{x})[x-\bar{x}, x-\bar{x}, x-\bar{x}] + ... + \frac{1}{k!}D^kf(\bar{x})[\underbrace{x-\bar{x}, ..., x-\bar{x}}_{k \text{ times}}] \end{array} \qquad (A.6.13)$$

We are already acquainted with the Taylor expansion of order 1

$$F_1(x) = f(\bar{x}) + Df(\bar{x})[x-\bar{x}]$$

– this is the affine function of x which approximates "very well" $f(x)$ in a neighbourhood of $\bar{x}$, namely, within approximation error $\bar{o}(|x-\bar{x}|)$. Similar fact is true for Taylor expansions of higher order:

Theorem A.6.4 *Let $f : \mathbf{R}^n \to \mathbf{R}^m$ be C^k in a neighbourhood of $\bar{x}$, and let $F_k(x)$ be the Taylor expansion of f at $\bar{x}$ of degree k. Then*

(i) *$F_k(x)$ is a vector-valued polynomial of full degree $\le k$ (i.e., every one of the coordinates of the vector $F_k(x)$ is a polynomial of $x_1, ..., x_n$, and the sum of powers of x_i's in every term of this polynomial does not exceed k);*

(ii) *$F_k(x)$ approximates $f(x)$ in a neighbourhood of $\bar{x}$ up to a remainder which is $\bar{o}(|x-\bar{x}|^k)$ as $x \to \bar{x}$:*

For every $\epsilon > 0$, there exists $\delta > 0$ such that

$$|x - \bar{x}| \le \delta \Rightarrow |F_k(x) - f(x)| \le \epsilon |x - \bar{x}|^k.$$

$F_k(\cdot)$ is the unique polynomial with components of full degree $\le k$ which approximates f up to a remainder which is $\bar{o}(|x - \bar{x}|^k)$.

(iii) *The value and the derivatives of F_k of orders $1, 2, ..., k$, taken at $\bar{x}$, are the same as the value and the corresponding derivatives of f taken at the same point.*

As stated in Theorem, $F_k(x)$ approximates $f(x)$ for x close to $\bar{x}$ up to a remainder which is $\bar{o}(|x - \bar{x}|^k)$. In many cases, it is not enough to know that the reminder is "$\bar{o}(|x - \bar{x}|^k)$" — we need an explicit bound on this remainder. The standard bound of this type is as follows:

Theorem A.6.5 *Let k be a positive integer, and let $f : \mathbf{R}^n \to \mathbf{R}^m$ be C^{k+1} in a ball $B_r = B_r(\bar{x}) = \{x \in \mathbf{R}^n : |x - \bar{x}| < r\}$ of a radius $r > 0$ centered at a point $\bar{x}$. Assume that the directional derivatives of order $k+1$, taken at every point of B_r along every unit direction, do not exceed certain $L < \infty$:*

$$|D^{k+1} f(x)[d, ..., d]| \le L \quad \forall (x \in B_r) \forall (d, |d| = 1).$$

Then for the Taylor expansion F_k of order k of f taken at $\bar{x}$ one has

$$|f(x) - F_k(x)| \le \frac{L|x - \bar{x}|^{k+1}}{(k+1)!} \quad \forall (x \in B_r).$$

Thus, in a neighbourhood of $\bar{x}$ the remainder of the k-th order Taylor expansion, taken at $\bar{x}$, is of order of $L|x - \bar{x}|^{k+1}$, where L is the maximal (over all unit directions and all points from the neighbourhood) magnitude of the directional derivatives of order $k+1$ of f.

A.7 Symmetric matrices

A.7.1 Spaces of matrices

Let $\mathbf{S}^m$ be the space of symmetric $m \times m$ matrices, and $\mathbf{M}^{m,n}$ be the space of rectangular $m \times n$ matrices with real entries. From the viewpoint of their linear structure (i.e., the operations of addition and multiplication by reals) $\mathbf{S}^m$ is just the arithmetic linear space $\mathbf{R}^{m(m+1)/2}$ of dimension $\frac{m(m+1)}{2}$: by arranging the elements of a symmetric $m \times m$ matrix X in a single column, say, in the row-by-row order, you get a usual m^2-dimensional column vector; multiplication of a matrix by a real and addition of matrices correspond to the same operations with the "representing vector(s)". When X runs through $\mathbf{S}^m$, the vector representing X runs through $m(m+1)/2$-dimensional subspace of $\mathbf{R}^{m^2}$ consisting of vectors satisfying the "symmetry condition" – the coordinates coming from symmetric to each other pairs of entries in X are equal to each other. Similarly, $\mathbf{M}^{m,n}$ as a linear space is just $\mathbf{R}^{mn}$, and it is natural to equip $\mathbf{M}^{m,n}$ with the inner product defined as the usual inner product of the vectors representing the matrices:

$$\langle X, Y \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} Y_{ij} = \mathrm{Tr}(X^T Y).$$

Here Tr stands for the *trace* – the sum of diagonal elements of a (square) matrix. With this inner product (called the *Frobenius inner product*), $\mathbf{M}^{m,n}$ becomes a legitimate Euclidean space, and we may use in

connection with this space all notions based upon the Euclidean structure, e.g., the (Frobenius) norm of a matrix

$$\| X \|_2 = \sqrt{\langle X, X \rangle} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}^2} = \sqrt{\mathrm{Tr}(X^T X)}$$

and likewise the notions of orthogonality, orthogonal complement of a linear subspace, etc. The same applies to the space $\mathbf{S}^m$ equipped with the Frobenius inner product; of course, the Frobenius inner product of symmetric matrices can be written without the transposition sign:

$$\langle X, Y \rangle = \mathrm{Tr}(XY), \ X, Y \in \mathbf{S}^m.$$

A.7.2 Main facts on symmetric matrices

Let us focus on the space $\mathbf{S}^m$ of symmetric matrices. The most important property of these matrices is as follows:

Theorem A.7.1 [Eigenvalue decomposition] *$n \times n$ matrix A is symmetric if and only if it admits an orthonormal system of eigenvectors: there exist orthonormal basis $\{e_1, ..., e_n\}$ such that*

$$Ae_i = \lambda_i e_i, \ i = 1, ..., n, \tag{A.7.1}$$

for reals λ_i.

In connection with Theorem A.7.1, it is worthy to recall the following notions and facts:

A.7.2.A. Eigenvectors and eigenvalues. An *eigenvector* of an $n \times n$ matrix A is a *nonzero* vector e (real or complex) such that $Ae = \lambda e$ for (real or complex) scalar λ; this scalar is called the *eigenvalue* of A *corresponding to the eigenvector* e.

Eigenvalues of A are exactly the roots of the *characteristic polynomial*

$$\pi(z) = \det(zI - A) = z^n + b_1 z^{n-1} + b_2 z^{n-2} + ... + b_n$$

of A.

Theorem A.7.1 states, in particular, that for a symmetric matrix A, all eigenvalues are real, and the corresponding eigenvectors can be chosen to be real and to form an orthonormal basis in $\mathbf{R}^n$.

A.7.2.B. Eigenvalue decomposition of a symmetric matrix. Theorem A.7.1 admits equivalent reformulation as follows (check the equivalence!):

Theorem A.7.2 *An $n \times n$ matrix A is symmetric if and only if it can be represented in the form*

$$A = U \Lambda U^T, \tag{A.7.2}$$

where

- *U is an orthogonal matrix: $U^{-1} = U^T$ (or, which is the same, $U^T U = I$, or, which is the same, $UU^T = I$, or, which is the same, the columns of U form an orthonormal basis in $\mathbf{R}^n$, or, which is the same, the columns of U form an orthonormal basis in $\mathbf{R}^n$).*
- *Λ is the diagonal matrix with the diagonal entries $\lambda_1, ..., \lambda_n$.*

Representation (A.7.2) with orthogonal U and diagonal Λ is called the *eigenvalue decomposition* of A. In such a representation,

- The columns of U form an orthonormal system of eigenvectors of A;
- The diagonal entries in Λ are the eigenvalues of A corresponding to these eigenvectors.

A.7.2.C. Vector of eigenvalues. When speaking about eigenvalues $\lambda_i(A)$ of a symmetric $n \times n$ matrix A, we always arrange them in the non-ascending order:

$$\lambda_1(A) \geq \lambda_2(A) \geq ... \geq \lambda_n(A);$$

$\lambda(A) \in \mathbf{R}^n$ denotes the vector of eigenvalues of A taken in the above order.

A.7.2.D. Freedom in eigenvalue decomposition. Part of the data Λ, U in the eigenvalue decomposition (A.7.2) is uniquely defined by A, while the other data admit certain "freedom". Specifically, the sequence $\lambda_1, ..., \lambda_n$ of eigenvalues of A (i.e., diagonal entries of Λ) is exactly the sequence of roots of the characteristic polynomial of A (every root is repeated according to its multiplicity) and thus is uniquely defined by A (provided that we arrange the entries of the sequence in the non-ascending order). The columns of U are not uniquely defined by A. What is uniquely defined, are the *linear spans* $E(\lambda)$ of the columns of U corresponding to all eigenvalues equal to certain λ; such a linear span is nothing but the *spectral subspace* $\{x : Ax = \lambda x\}$ of A corresponding to the eigenvalue λ. There are as many spectral subspaces as many different eigenvalues; spectral subspaces corresponding to different eigenvalues of symmetric matrix are orthogonal to each other, and their sum is the entire space. When building an orthogonal matrix U in the spectral decomposition, one chooses an orthonormal eigenbasis in the spectral subspace corresponding to the largest eigenvalue and makes the vectors of this basis the first columns in U, then chooses an orthonormal basis in the spectral subspace corresponding to the second largest eigenvalue and makes the vector from this basis the next columns of U, and so on.

A.7.2.E. "Simultaneous" decomposition of commuting symmetric matrices. Let $A_1, ..., A_k$ be $n \times n$ symmetric matrices. It turns out that *the matrices commute with each other ($A_iA_j = A_jA_i$ for all i, j) if and only if they can be "simultaneously diagonalized", i.e., there exist a single orthogonal matrix U and diagonal matrices Λ_1,...,Λ_k such that*

$$A_i = U\Lambda_i U^T,\ i = 1, ..., k.$$

You are welcome to prove this statement by yourself; to simplify your task, here are two simple and important by their own right statements which help to reach your target:

> **A.7.2.E.1:** *Let λ be a real and A, B be two commuting $n \times n$ matrices. Then the spectral subspace $E = \{x : Ax = \lambda x\}$ of A corresponding to λ is invariant for B (i.e., $Be \in E$ for every $e \in E$).*
>
> **A.7.2.E.2:** *If A is an $n \times n$ matrix and L is an invariant subspace of A (i.e., L is a linear subspace such that $Ae \in L$ whenever $e \in L$), then the orthogonal complement $L^\perp$ of L is invariant for the matrix A^T. In particular, if A is symmetric and L is invariant subspace of A, then $L^\perp$ is invariant subspace of A as well.*

A.7.3 Variational characterization of eigenvalues

Theorem A.7.3 [VCE – Variational Characterization of Eigenvalues] *Let A be a symmetric matrix. Then*

$$\lambda_\ell(A) = \min_{E \in \mathcal{E}_\ell} \max_{x \in E, x^Tx=1} x^TAx,\ \ell = 1, ..., n, \tag{A.7.3}$$

where $\mathcal{E}_\ell$ is the family of all linear subspaces in $\mathbf{R}^n$ of the dimension $n - \ell + 1$.

VCE says that to get the largest eigenvalue $\lambda_1(A)$, you should maximize the quadratic form x^TAx over the unit sphere $S = \{x \in \mathbf{R}^n : x^Tx = 1\}$; the maximum is exactly $\lambda_1(A)$. To get the second largest eigenvalue $\lambda_2(A)$, you should act as follows: you choose a linear subspace E of dimension $n - 1$ and maximize the quadratic form x^TAx over the cross-section of S by this subspace; the maximum value of the form depends on E, and you minimize this maximum over linear subspaces E of the dimension

$n-1$; the result is exactly $\lambda_2(A)$. To get $\lambda_3(A)$, you replace in the latter construction subspaces of the dimension $n-1$ by those of the dimension $n-2$, and so on. In particular, the smallest eigenvalue $\lambda_n(A)$ is just the minimum, over all linear subspaces E of the dimension $n-n+1=1$, i.e., over all lines passing through the origin, of the quantities x^TAx, where $x \in E$ is unit ($x^Tx=1$); in other words, $\lambda_n(A)$ is just the minimum of the quadratic form x^TAx over the unit sphere S.

Proof of the VCE is pretty easy. Let $e_1,...,e_n$ be an orthonormal eigenbasis of A: $Ae_\ell = \lambda_\ell(A)e_\ell$. For $1 \le \ell \le n$, let $F_\ell = \mathrm{Lin}\{e_1,...,e_\ell\}$, $G_\ell = \mathrm{Lin}\{e_\ell, e_{\ell+1},...,e_n\}$. Finally, for $x \in \mathbf{R}^n$ let $\xi(x)$ be the vector of coordinates of x in the orthonormal basis $e_1,...,e_n$. Note that

$$x^Tx = \xi^T(x)\xi(x),$$

since $\{e_1,...,e_n\}$ is an orthonormal basis, and that

$$\begin{aligned} x^TAx &= x^TA\sum_i \xi_i(x)e_i) = x^T\sum_i \lambda_i(A)\xi_i(x)e_i = \\ \sum_i \lambda_i(A)\xi_i(x)\underbrace{(x^Te_i)}_{\xi_i(x)} & \\ &= \sum_i \lambda_i(A)\xi_i^2(x). \end{aligned} \tag{A.7.4}$$

Now, given ℓ, $1 \le \ell \le n$, let us set $E = G_\ell$; note that E is a linear subspace of the dimension $n-\ell+1$. In view of (A.7.4), the maximum of the quadratic form x^TAx over the intersection of our E with the unit sphere is

$$\max\left\{\sum_{i=\ell}^n \lambda_i(A)\xi_i^2 : \sum_{i=\ell}^n \xi_i^2 = 1\right\},$$

and the latter quantity clearly equals to $\max\limits_{\ell \le i \le n} \lambda_i(A) = \lambda_\ell(A)$. Thus, for appropriately chosen $E \in \mathcal{E}_\ell$, the inner maximum in the right hand side of (A.7.3) equals to $\lambda_\ell(A)$, whence the right hand side of (A.7.3) is $\le \lambda_\ell(A)$. It remains to prove the opposite inequality. To this end, consider a linear subspace E of the dimension $n-\ell+1$ and observe that it has nontrivial intersection with the linear subspace F_ℓ of the dimension ℓ (indeed, $\dim E + \dim F_\ell = (n-\ell+1)+\ell > n$, so that $\dim(E \cap F) > 0$ by the Dimension formula). It follows that there exists a unit vector y belonging to both E and F_ℓ. Since y is a unit vector from F_ℓ, we have $y = \sum\limits_{i=1}^{\ell} \eta_i e_i$ with $\sum\limits_{i=1}^{\ell} \eta_i^2 = 1$, whence, by (A.7.4),

$$y^TAy = \sum_{i=1}^{\ell} \lambda_i(A)\eta_i^2 \ge \min_{1 \le i \le \ell} \lambda_i(A) = \lambda_\ell(A).$$

Since y is in E, we conclude that

$$\max_{x \in E: x^Tx=1} x^TAx \ge y^TAy \ge \lambda_\ell(A).$$

Since E is an arbitrary subspace form $\mathcal{E}_\ell$, we conclude that the right hand side in (A.7.3) is $\ge \lambda_\ell(A)$. ∎

A simple and useful byproduct of our reasoning is the relation (A.7.4):

Corollary A.7.1 *For a symmetric matrix A, the quadratic form x^TAx is weighted sum of squares of the coordinates $\xi_i(x)$ of x taken with respect to an orthonormal eigenbasis of A; the weights in this sum are exactly the eigenvalues of A:*

$$x^TAx = \sum_i \lambda_i(A)\xi_i^2(x).$$

A.7.3.1 Corollaries of the VCE

VCE admits a number of extremely important corollaries as follows:

A.7.3.A. Eigenvalue characterization of positive (semi)definite matrices. Recall that a matrix A is called positive definite (notation: $A \succ 0$), if it is symmetric and the quadratic form x^TAx is positive outside the origin; A is called positive semidefinite (notation: $A \succeq 0$), if A is symmetric and the quadratic form x^TAx is nonnegative everywhere. VCE provides us with the following eigenvalue characterization of positive (semi)definite matrices:

Proposition A.7.1 *: A symmetric matrix A is positive semidefinite if and only if its eigenvalues are nonnegative; A is positive definite if and only if all eigenvalues of A are positive*

Indeed, A is positive definite, if and only if the minimum value of x^TAx over the unit sphere is positive, and is positive semidefinite, if and only if this minimum value is nonnegative; it remains to note that by VCE, the minimum value of x^TAx over the unit sphere is exactly the minimum eigenvalue of A.

A.7.3.B. $\succeq$-Monotonicity of the vector of eigenvalues. Let us write $A \succeq B$ ($A \succ B$) to express that A, B are symmetric matrices of the same size such that $A - B$ is positive semidefinite (respectively, positive definite).

Proposition A.7.2 *If $A \succeq B$, then $\lambda(A) \geq \lambda(B)$, and if $A \succ B$, then $\lambda(A) > \lambda(B)$.*

Indeed, when $A \succeq B$, then, of course,

$$\max_{x \in E: x^Tx=1} x^TAx \geq \max_{x \in E: x^Tx=1} x^TBx$$

for every linear subspace E, whence

$$\lambda_\ell(A) = \min_{E \in \mathcal{E}_\ell} \max_{x \in E: x^Tx=1} x^TAx \geq \min_{E \in \mathcal{E}_\ell} \max_{x \in E: x^Tx=1} x^TBx = \lambda_\ell(B),\ \ell = 1, ..., n,$$

i.e., $\lambda(A) \geq \lambda(B)$. The case of $A \succ B$ can be considered similarly.

A.7.3.C. Eigenvalue Interlacement Theorem. We shall formulate this extremely important theorem as follows:

Theorem A.7.4 [Eigenvalue Interlacement Theorem] *Let A be a symmetric $n \times n$ matrix and $\bar{A}$ be the angular $(n-k) \times (n-k)$ submatrix of A. Then, for every $\ell \leq n - k$, the ℓ-th eigenvalue of $\bar{A}$ separates the ℓ-th and the $(\ell + k)$-th eigenvalues of A:*

$$\lambda_\ell(A) \succeq \lambda_\ell(\bar{A}) \succeq \lambda_{\ell+k}(A). \tag{A.7.5}$$

Indeed, by VCE, $\lambda_\ell(\bar{A}) = \min_{E \in \bar{\mathcal{E}}_\ell} \max_{x \in E: x^Tx=1} x^TAx$, where $\bar{\mathcal{E}}_\ell$ is the family of all linear subspaces of the dimension $n - k - \ell + 1$ contained in the linear subspace $\{x \in \mathbf{R}^n : x_{n-k+1} = x_{n-k+2} = ... = x_n = 0\}$. Since $\bar{\mathcal{E}}_\ell \subset \mathcal{E}_{\ell+k}$, we have

$$\lambda_\ell(\bar{A}) = \min_{E \in \bar{\mathcal{E}}_\ell} \max_{x \in E: x^Tx=1} x^TAx \geq \min_{E \in \mathcal{E}_{\ell+k}} \max_{x \in E: x^Tx=1} x^TAx = \lambda_{\ell+k}(A).$$

We have proved the left inequality in (A.7.5). Applying this inequality to the matrix $-A$, we get

$$-\lambda_\ell(\bar{A}) = \lambda_{n-k-\ell}(-\bar{A}) \geq \lambda_{n-\ell}(-A) = -\lambda_\ell(A),$$

or, which is the same, $\lambda_\ell(\bar{A}) \leq \lambda_\ell(A)$, which is the first inequality in (A.7.5).

A.7.4 Positive semidefinite matrices and the semidefinite cone

A.7.4.A. Positive semidefinite matrices. Recall that an $n \times n$ matrix A is called *positive semidefinite* (notation: $A \succeq 0$), if A is symmetric and produces nonnegative quadratic form:

$$A \succeq 0 \Leftrightarrow \{A = A^T \quad \text{and} \quad x^T Ax \geq 0 \quad \forall x\}.$$

A is called positive *definite* (notation: $A \succ 0$), if it is positive semidefinite and the corresponding quadratic form is positive outside the origin:

$$A \succ 0 \Leftrightarrow \{A = A^T \quad \text{and} \quad x^T Ax > 00 \quad \forall x \neq 0\}.$$

It makes sense to list a number of equivalent definitions of a positive semidefinite matrix:

Theorem A.7.5 *Let A be a symmetric $n \times n$ matrix. Then the following properties of A are equivalent to each other:*

(i) $A \succeq 0$
(ii) $\lambda(A) \geq 0$
(iii) $A = D^T D$ *for certain rectangular matrix* D
(iv) $A = \Delta^T \Delta$ *for certain upper triangular* $n \times n$ *matrix* Δ
(v) $A = B^2$ *for certain symmetric matrix* B;
(vi) $A = B^2$ *for certain* $B \succeq 0$.

The following properties of a symmetric matrix A also are equivalent to each other:

(i′) $A \succ 0$
(ii′) $\lambda(A) > 0$
(iii′) $A = D^T D$ *for certain rectangular matrix* D *of rank* n
(iv′) $A = \Delta^T \Delta$ *for certain nondegenerate upper triangular* $n \times n$ *matrix* Δ
(v′) $A = B^2$ *for certain nondegenerate symmetric matrix* B;
(vi′) $A = B^2$ *for certain* $B \succ 0$.

Proof. (i)⇔(ii): this equivalence is stated by Proposition A.7.1.

(ii)⇔(vi): Let $A = U\Lambda U^T$ be the eigenvalue decomposition of A, so that U is orthogonal and Λ is diagonal with nonnegative diagonal entries $\lambda_i(A)$ (we are in the situation of (ii) !). Let $\Lambda^{1/2}$ be the diagonal matrix with the diagonal entries $\lambda_i^{1/2}(A)$; note that $(\Lambda^{1/2})^2 = \Lambda$. The matrix $B = U\Lambda^{1/2}U^T$ is symmetric with nonnegative eigenvalues $\lambda_i^{1/2}(A)$, so that $B \succeq 0$ by Proposition A.7.1, and

$$B^2 = U\Lambda^{1/2}\underbrace{U^T U}_{I}\Lambda^{1/2}U^T = U(\Lambda^{1/2})^2 U^T = U\Lambda U^T = A,$$

as required in (vi).

(vi)⇒(v): evident.

(v)⇒(iv): Let $A = B^2$ with certain symmetric B, and let b_i be i-th column of B. Applying the Gram-Schmidt orthogonalization process (see proof of Theorem A.2.3.(iii)), we can find an orthonormal system of vectors $u_1, ..., u_n$ and lower triangular matrix L such that $b_i = \sum_{j=1}^{i} L_{ij}u_j$, or, which is the same, $B^T = LU$, where U is the orthogonal matrix with the rows $u_1^T, ..., u_n^T$. We now have $A = B^2 = B^T(B^T)^T = LUU^T L^T = LL^T$. We see that $A = \Delta^T\Delta$, where the matrix $\Delta = L^T$ is upper triangular.

(iv)⇒(iii): evident.

(iii)⇒(i): If $A = D^T D$, then $x^T Ax = (Dx)^T(Dx) \geq 0$ for all x.

We have proved the equivalence of the properties (i) – (vi). Slightly modifying the reasoning (do it yourself!), one can prove the equivalence of the properties (i′) – (vi′). ∎

Remark A.7.1 (i) [Checking positive semidefiniteness] Given an $n \times n$ symmetric matrix A, one can check whether it is positive semidefinite by a purely algebraic finite algorithm (the so called *Lagrange diagonalization of a quadratic form*) which requires at most $O(n^3)$ arithmetic operations. Positive definiteness

of a matrix can be checked also by the *Choleski factorization algorithm* which finds the decomposition in (iv′), if it exists, in approximately $\frac{1}{6}n^3$ arithmetic operations.

There exists another useful algebraic criterion (Sylvester's criterion) for positive semidefiniteness of a matrix; according to this criterion, a symmetric matrix A is positive definite if and only if its angular minors are positive, and A is positive semidefinite if and only if all its principal minors are nonnegative. For example, a symmetric 2×2 matrix $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ is positive semidefinite if and only if $a \geq 0$, $c \geq 0$ and $\det(A) \equiv ac - b^2 \geq 0$.

(ii) [Square root of a positive semidefinite matrix] By the first chain of equivalences in Theorem A.7.5, a symmetric matrix A is $\succeq 0$ if and only if A is the square of a positive semidefinite matrix B. The latter matrix is uniquely defined by $A \succeq 0$ and is called the *square root* of A (notation: $A^{1/2}$).

A.7.4.B. The semidefinite cone. When adding symmetric matrices and multiplying them by reals, we add, respectively multiply by reals, the corresponding quadratic forms. It follows that

> **A.7.4.B.1:** *The sum of positive semidefinite matrices and a product of a positive semidefinite matrix and a nonnegative real is positive semidefinite*,

or, which is the same (see Section 1.1.4),

> **A.7.4.B.2:** *$n \times n$ positive semidefinite matrices form a cone $\mathbf{S}^n_+$ in the Euclidean space $\mathbf{S}^n$ of symmetric $n \times n$ matrices, the Euclidean structure being given by the Frobenius inner product $\langle A, B\rangle = \mathrm{Tr}(AB) = \sum_{i,j} A_{ij}B_{ij}$.*

The cone $\mathbf{S}^n_+$ is called the *semidefinite* cone of size n. It is immediately seen that the semidefinite cone $\mathbf{S}^n_+$ is "good" (see Lecture 5), specifically,

- $\mathbf{S}^n_+$ is closed: the limit of a converging sequence of positive semidefinite matrices is positive semidefinite;
- $\mathbf{S}^n_+$ is pointed: the only $n \times n$ matrix A such that both A and $-A$ are positive semidefinite is the zero $n \times n$ matrix;
- $\mathbf{S}^n_+$ possesses a nonempty interior which is comprised of positive definite matrices.

Note that the relation $A \succeq B$ means exactly that $A - B \in \mathbf{S}^n_+$, while $A \succ B$ is equivalent to $A - B \in \mathrm{int}\, \mathbf{S}^n_+$. The "matrix inequalities" $A \succeq B$ ($A \succ B$) match the standard properties of the usual scalar inequalities, e.g.:

$A \succeq A$	[reflexivity]
$A \succeq B$, $B \succeq A \Rightarrow A = B$	[antisymmetry]
$A \succeq B$, $B \succeq C \Rightarrow A \succeq C$	[transitivity]
$A \succeq B$, $C \succeq D \Rightarrow A + C \succeq B + D$	[compatibility with linear operations, I]
$A \succeq B$, $\lambda \geq 0 \Rightarrow \lambda A \succeq \lambda B$	[compatibility with linear operations, II]
$A_i \succeq B_i$, $A_i \to A$, $B_i \to B$ as $i \to \infty \Rightarrow A \succeq B$	[closedness]

with evident modifications when $\succeq$ is replaced with $\succ$, or

$$A \succeq B, C \succ D \Rightarrow A + C \succ B + D,$$

etc. Along with these standard properties of inequalities, the inequality $\succeq$ possesses a nice additional property:

> **A.7.4.B.3:** *In a valid $\succeq$-inequality*
>
> $$A \succeq B$$
>
> *one can multiply both sides from the left and by the right by a (rectangular) matrix and its transpose:*
>
> $$A, B \in \mathbf{S}^n, \quad A \succeq B, \quad V \in \mathbf{M}^{n,m}$$
> $$\Downarrow$$
> $$V^T A V \succeq V^T B V$$

Indeed, we should prove that if $A - B \succeq 0$, then also $V^T(A-B)V \succeq 0$, which is immediate – the quadratic form $y^T[V^T(A-B)V]y = (Vy)^T(A-B)(Vy)$ of y is nonnegative along with the quadratic form $x^T(A-B)x$ of x.

An important additional property of the semidefinite cone is its *self-duality*:

Theorem A.7.6 *A symmetric matrix Y has nonnegative Frobenius inner products with all positive semidefinite matrices if and only if Y itself is positive semidefinite.*

Proof. "if" part: Assume that $Y \succeq 0$, and let us prove that then $\mathrm{Tr}(YX) \geq 0$ for every $X \succeq 0$. Indeed, the eigenvalue decomposition of Y can be written as

$$Y = \sum_{i=1}^{n} \lambda_i(Y) e_i e_i^T,$$

where e_i are the orthonormal eigenvectors of Y. We now have

$$\begin{array}{rcl} \mathrm{Tr}(YX) & = & \mathrm{Tr}((\sum_{i=1}^{n} \lambda_i(Y) e_i e_i^T)X) = \sum_{i=1}^{n} \lambda_i(Y)\,\mathrm{Tr}(e_i e_i^T X) \\ & = & \sum_{i=1}^{n} \lambda_i(Y)\,\mathrm{Tr}(e_i^T X e_i), \end{array} \tag{A.7.6}$$

where the concluding equality is given by the following well-known property of the trace:

A.7.4.B.4: *Whenever matrices A, B are such that the product AB makes sense and is a square matrix, one has*

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA).$$

Indeed, we should verify that if $A \in \mathbf{M}^{p,q}$ and $B \in \mathbf{M}^{q,p}$, then $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$. The left hand side quantity in our hypothetic equality is $\sum_{i=1}^{p}\sum_{j=1}^{q} A_{ij}B_{ji}$, and the right hand side quantity is $\sum_{j=1}^{q}\sum_{i=1}^{p} B_{ji}A_{ij}$; they indeed are equal.

Looking at the concluding quantity in (A.7.6), we see that it indeed is nonnegative whenever $X \succeq 0$ (since $Y \succeq 0$ and thus $\lambda_i(Y) \geq 0$ by P.7.5).

"only if" part: We are given Y such that $\mathrm{Tr}(YX) \geq 0$ for all matrices $X \succeq 0$, and we should prove that $Y \succeq 0$. This is immediate: for every vector x, the matrix $X = xx^T$ is positive semidefinite (Theorem A.7.5.(iii)), so that $0 \leq \mathrm{Tr}(Yxx^T) = \mathrm{Tr}(x^TYx) = x^TYx$. Since the resulting inequality $x^TYx \geq 0$ is valid for every x, we have $Y \succeq 0$. ∎

A.7.4.B.5: Schur Complement Lemma is the following simple and extremely useful fact:

Proposition A.7.3 [Schur Complement Lemma] *A symmetric block 2×2 matrix*

$$A = \left[\begin{array}{c|c} P & Q \\ \hline Q^T & R \end{array}\right]$$

with $R \succ 0$ is positive definite (semidefinite) if and only if the matrix

$$P - QR^{-1}Q^T$$

is positive definite (resp. semidefinite).

Proof is immediate. When $P \in \mathbf{S}^p$, $R \in \mathbf{S}^q$, splitting vector x from $\mathbf{R}^{p+q}$ into blocks $u \in \mathbf{R}^p$, $v \in \mathbf{R}^q$,

$$\begin{array}{rl} & P \succeq 0 \Leftrightarrow u^TPu + 2u^TQv + v^TRv \geq 0\ \forall (u,v) \\ \Leftrightarrow & \min_v \left[v^TRv + 2v^TQu + u^TPu\right] \geq 0\,\forall u \\ \Leftrightarrow & \min_c \left[(v - R^{-1}Qu)^TR(v - R^{-1}Q)u + u^TPu - u^TQR^{-1}Q^Tu\right] \geq 0\,\forall u \\ \Leftrightarrow & P - QR^{-1}Q^T \succeq 0, \end{array}$$

as claimed by the "positive semidefinite" verion of Lemma. The same reasoning, with evident modifications, justifies the "positive definite" claim. ∎