J. Frédéric Bonnans, J. Charles Gilbert
Claude Lemaréchal, Claudia A. Sagastizábal

Numerical Optimization
Theoretical and Practical Aspects

May 31, 2006

Springer
Berlin Heidelberg NewYork
Barcelona Hong Kong
London Milan Paris
Tokyo

Preface

This book is entirely devoted to numerical algorithms for optimization, their theoretical foundations and convergence properties, as well as their implementation, their use, and other practical aspects. The aim is to familiarize the reader with these numerical algorithms: understanding their behaviour in practice, properly using existing software libraries, adequately designing and implementing "home-made" methods, correctly diagnosing the causes of possible difficulties. Expected readers are engineers, Master or Ph.D. students, confirmed researchers, in applied mathematics or from various other disciplines where optimization is a need.

Our aim is therefore not to give most accurate results in optimization, nor to detail the latest refinements of such and such method. First of all, little is said concerning optimization theory itself (optimality conditions, constraint qualification, stability theory). As for algorithms, we limit ourselves most of the time to stable and well-established material. Throughout we keep as a leading thread the actual *practical value* of optimization methods, in terms of their efficiency to solve real-world problems. Nevertheless, serious attention is paid to the theoretical properties of optimization methods: this book is mainly based upon theorems. Besides, some new and promising results or approaches could not be completely discarded; they are also presented, generally in the form of special sections, mainly aimed at orienting the reader to the relevant bibliography.

An introductory chapter gives some generalities on optimization and iterative algorithms. It contains in particular motivating examples, ranking from meteorological forecast to power production management; they illustrate the large field of branches where optimization finds its applications. Then come four parts, rather independent of each other. The first one is devoted to algorithms for unconstrained optimization which, in addition to their direct usefulness, are a basis for more complex problems. The second part concerns rather special methods, applicable when the usual differentiability assumptions are not satisfied. Such methods appear in the decomposition of large-scale problems and the relaxation of combinatorial problems. Nonlinearly constrained optimization forms the third part, substantially more technical, as the subject is still in evolution. Finally, the fourth part gives a deep account of the more recent interior point methods, originally designed for the simpler problems of linear and quadratic programming, and whose application to more general situations is the subject of active research.

This book is a translated and improved version of the monograph [43], written in French. The French monograph was used as the textbook of an intensive two week course given several times by the authors, both in France and abroad. Each topic was presented from a theoretical point of view in morning lectures. The afternoons were devoted to implementation issues and related computational work. The conception of such a course is due to J.-B. Hiriart-Urruty, to whom the authors are deeply indebted.

Finally, three of the authors express their warm gratitude to Claude Lemaréchal for having given the impetus to this new work by providing a first English version.

Notes on this revised edition. Besides minor corrections, the present version contains substantial changes with respect to the first edition. First of all, (simplified but) nontrivial application problems have been inserted. They involve the typical operations to be performed when one is faced with a real-life application: modelling, choice of methodology and some theoretical work to motivate it, computer implementation. Such computational exercises help getting a better understanding of optimization methods beyond their theoretical description, by addressing important features to be taken into account when passing to implementation of any numerical algorithm.

In addition, the theoretical background in Part I now includes a discussion on global convergence, and a section on the classical pivotal approach to quadratic programming. Part II has been completely reorganized and expanded. The introductory chapter, on basic subdifferential calculus and duality theory, has two examples of nonsmooth functions that appear often in practice and serve as motivation (pointwise maximum and dual functions). A new section on convergence results for bundle methods has been added. The chapter on applications of nonsmooth optimization, previously focusing on decomposition of complex problems via Lagrangian duality, describes also extensions of bundle methods for handling varying dimensions, for solving constrained problems, and for solving generalized equations. Also, a brief commented review of existing software for nonlinear optimization has been added in Part III.

Finally, the reader will find additional information at `http://www-rocq.inria.fr/~gilbert/bgls`. The page gathers the data for running the test problems, various optimization codes, including an SQP solver (in Matlab), and pieces of software that solve the computational exercises.

Paris, Grenoble, Rio de Janeiro,
May 2006

J. Frédéric Bonnans
J. Charles Gilbert
Claude Lemaréchal
Claudia A. Sagastizábal

Table of Contents

Preliminaries

1 General Introduction 3
1.1 Generalities on Optimization 3
1.1.1 The Problem 3
1.1.2 Classification 4
1.2 Motivation and Examples 5
1.2.1 Molecular Biology 5
1.2.2 Meteorology 6
1.2.3 Trajectory of a Deepwater Vehicle 8
1.2.4 Optimization of Power Management 9
1.3 General Principles of Resolution 10
1.4 Convergence: Global Aspects 12
1.5 Convergence: Local Aspects 14
1.6 Computing the Gradient 16
Bibliographical Comments 19

Part I Unconstrained Problems

2 Basic Methods 25
2.1 Existence Questions 25
2.2 Optimality Conditions 26
2.3 First-Order Methods 27
2.3.1 Gauss-Seidel 27
2.3.2 Method of Successive Approximations, or Gradient Method 28
2.4 Link with the General Descent Scheme 28
2.4.1 Choosing the ℓ_1-Norm 29
2.4.2 Choosing the ℓ_2-Norm 30
2.5 Steepest-Descent Method 30
2.6 Implementation 34
Bibliographical Comments 35

3 Line-Searches 37
3.1 General Scheme 37
3.2 Computing the New t 40
3.3 Optimal Stepsize (for the record only) 42
3.4 Modern Line-Search: Wolfe's Rule 43
3.5 Other Line-Searches: Goldstein and Price, Armijo 47
3.5.1 Goldstein and Price 47
3.5.2 Armijo 47
3.5.3 Remark on the Choice of Constants 48
3.6 Implementation Considerations 49
Bibliographical Comments 50

4 Newtonian Methods 51
4.1 Preliminaries 51
4.2 Forcing Global Convergence 52
4.3 Alleviating the Method 53
4.4 Quasi-Newton Methods 54
4.5 Global Convergence 57
4.6 Local Convergence: Generalities 59
4.7 Local Convergence: BFGS 61
Bibliographical Comments 65

5 Conjugate Gradient 67
5.1 Outline of Conjugate Gradient 67
5.2 Developing the Method 69
5.3 Computing the Direction 70
5.4 The Algorithm Seen as an Orthogonalization Process 70
5.5 Application to Non-Quadratic Functions 72
5.6 Relation with Quasi-Newton 74
Bibliographical Comments 75

6 Special Methods 77
6.1 Trust-Regions 77
6.1.1 The Elementary Problem 78
6.1.2 The Elementary Mechanism: Curvilinear Search 79
6.1.3 Incidence on the Sequence x_k 81
6.2 Least-Squares Problems: Gauss-Newton 82
6.3 Large-Scale Problems: Limited-Memory Quasi-Newton 84
6.4 Truncated Newton 86
6.5 Quadratic Programming 88
6.5.1 The basic mechanism 89
6.5.2 The solution algorithm 90
6.5.3 Convergence 92
Bibliographical Comments 95

7 A Case Study: Seismic Reflection Tomography 97
7.1 Modelling 97
7.2 Computation of the Reflection Points 99
7.3 Gradient of the Traveltime 100
7.4 The Least-Squares Problem to Solve 101
7.5 Solving the Seismic Reflection Tomography Problem 102
General Conclusion 103

Part II Nonsmooth Optimization

8 Introduction to Nonsmooth Optimization 109
8.1 First Elements of Convex Analysis 109
8.2 Lagrangian Relaxation and Duality 111
8.2.1 Primal-Dual Relations 111
8.2.2 Back to the Primal. Recovering Primal Solutions 113
8.3 Two Convex Nondifferentiable Functions 116
8.3.1 Finite Minimax Problems 116
8.3.2 Dual Functions in Lagrangian Duality 117

9 Some Methods in Nonsmooth Optimization 119
9.1 Why Special Methods? 119
9.2 Descent Methods 120
9.2.1 Steepest-Descent Method 121
9.2.2 Stabilization. A Dual Approach. The ε-subdifferential 124
9.3 Two Black-Box Methods 126
9.3.1 Subgradient Methods 127
9.3.2 Cutting-Planes Method 130

10 Bundle Methods. The Quest for Descent 137
10.1 Stabilization. A Primal Approach 137
10.2 Some Examples of Stabilized Problems 140
10.3 Penalized Bundle Methods 141
10.3.1 A Trip to the Dual Space 144
10.3.2 Managing the Bundle. Aggregation 147
10.3.3 Updating the Penalization Parameter. Reversal Forms 150
10.3.4 Convergence Analysis 154

11 Applications of Nonsmooth Optimization 161
11.1 Divide to conquer. Decomposition methods 161
11.1.1 Price Decomposition 163
11.1.2 Resource Decomposition 167
11.1.3 Variable Partitioning or Benders Decomposition 169
11.1.4 Other Decomposition Methods 171

11.2 Transpassing Frontiers 172
11.2.1 Dynamic Bundle Methods 173
11.2.2 Constrained Bundle Methods 177
11.2.3 Bundle Methods for Generalized Equations 180

12 Computational Exercises 183
12.1 Building Prototypical NSO Black Boxes 183
12.1.1 The Function MAXQUAD 183
12.1.2 The Function MAXANAL 184
12.2 Implementation of Some NSO Methods 185
12.3 Running the Codes 186
12.4 Improving the Bundle Implementation 187
12.5 Decomposition Application 187

Part III Newton's Methods in Constrained Optimization

13 Background 197
13.1 Differential Calculus 197
13.2 Existence and Uniqueness of Solutions 199
13.3 First-Order Optimality Conditions 200
13.4 Second-Order Optimality Conditions 202
13.5 Speed of Convergence 203
13.6 Projection onto a Closed Convex Set 205
13.7 The Newton Method 205
13.8 The Hanging Chain Project I 208
Notes 213
Exercises 214

14 Local Methods for Problems with Equality Constraints . . 215
14.1 Newton's Method 216
14.2 Adapted Decompositions of $\mathbb{R}^n$ 222
14.3 Local Analysis of Newton's Method 227
14.4 Computation of the Newton Step 230
14.5 Reduced Hessian Algorithm 235
14.6 A Comparison of the Algorithms 243
14.7 The Hanging Chain Project II 245
Notes 250
Exercises 251

15 Local Methods for Problems with Equality and Inequality Constraints 255
15.1 The SQP Algorithm 256
15.2 Primal-Dual Quadratic Convergence 259
15.3 Primal Superlinear Convergence 264

15.4 The Hanging Chain Project III 267
Notes 270
Exercise 270

16 Exact Penalization 271
16.1 Overview 271
16.2 The Lagrangian 274
16.3 The Augmented Lagrangian 275
16.4 Nondifferentiable Augmented Function 279
Notes 284
Exercises 285

17 Globalization by Line-Search 289
17.1 Line-Search SQP Algorithms 291
17.2 Truncated SQP 298
17.3 From Global to Local 307
17.4 The Hanging Chain Project IV 316
Notes 320
Exercises 321

18 Quasi-Newton Versions 323
18.1 Principles 323
18.2 Quasi-Newton SQP 327
18.3 Reduced Quasi-Newton Algorithm 331
18.4 The Hanging Chain Project V 340

Part IV Interior-Point Algorithms for Linear and Quadratic Optimization

19 Linearly Constrained Optimization and Simplex Algorithm 353
19.1 Existence of Solutions 353
19.1.1 Existence Result 353
19.1.2 Basic Points and Extensions 355
19.2 Duality 356
19.2.1 Introducing the Dual Problem 357
19.2.2 Concept of Saddle-Point 358
19.2.3 Other Formulations 362
19.2.4 Strict Complementarity 363
19.3 The Simplex Algorithm 364
19.3.1 Computing the Descent Direction 364
19.3.2 Stating the algorithm 365
19.3.3 Dual simplex 367
19.4 Comments 368

20 Linear Monotone Complementarity and Associated Vector Fields . . . 371
20.1 Logarithmic Penalty and Central Path . . . 371
20.1.1 Logarithmic Penalty . . . 371
20.1.2 Central Path . . . 372
20.2 Linear Monotone Complementarity . . . 373
20.2.1 General Framework . . . 374
20.2.2 A Group of Transformations . . . 377
20.2.3 Standard Form . . . 378
20.2.4 Partition of Variables and Canonical Form . . . 379
20.2.5 Magnitudes in a Neighborhood of the Central Path . . 380
20.3 Vector Fields Associated with the Central Path . . . 382
20.3.1 General Framework . . . 383
20.3.2 Scaling the Problem . . . 383
20.3.3 Analysis of the Directions . . . 384
20.3.4 Modified Field . . . 387
20.4 Continuous Trajectories . . . 389
20.4.1 Limit Points of Continuous Trajectories . . . 389
20.4.2 Developing Affine Trajectories and Directions . . . 391
20.4.3 Mizuno's Lemma . . . 393
20.5 Comments . . . 393

21 Predictor-Corrector Algorithms . . . 395
21.1 Overview . . . 395
21.2 Statement of the Methods . . . 396
21.2.1 General Framework for Primal-Dual Algorithms . . . 396
21.2.2 Weighting After Displacement . . . 397
21.2.3 The Predictor-Corrector Method . . . 397
21.3 A small-Neighborhood Algorithm . . . 398
21.3.1 Statement of the Algorithm. Main Result . . . 398
21.3.2 Analysis of the Centralization Move . . . 398
21.3.3 Analysis of the Affine Step and Global Convergence . 399
21.3.4 Asymptotic Speed of Convergence . . . 401
21.4 A Predictor-Corrector Algorithm with Modified Field . . . 402
21.4.1 Principle . . . 402
21.4.2 Statement of the Algorithm. Main Result . . . 404
21.4.3 Complexity Analysis . . . 404
21.4.4 Asymptotic Analysis . . . 405
21.5 A Large-Neighborhood Algorithm . . . 406
21.5.1 Statement of the Algorithm. Main Result . . . 406
21.5.2 Analysis of the Centering Step . . . 407
21.5.3 Analysis of the Affine Step . . . 408
21.5.4 Asymptotic Convergence . . . 408
21.6 Practical Aspects . . . 408
21.7 Comments . . . 409

22 Non-Feasible Algorithms . . . 411
22.1 Overview . . . 411
22.2 Principle of the Non-Feasible Path Following . . . 411
22.2.1 Non-Feasible Central Path . . . 411
22.2.2 Directions of Move . . . 412
22.2.3 Orders of Magnitude of Approximately Centered Points . . . 413
22.2.4 Analysis of Directions . . . 415
22.2.5 Modified Field . . . 418
22.3 Non-Feasible Predictor-Corrector Algorithm . . . 419
22.3.1 Complexity Analysis . . . 420
22.3.2 Asymptotic Analysis . . . 422
22.4 Comments . . . 422

23 Self-Duality . . . 425
23.1 Overview . . . 425
23.2 Linear Problems with Inequality Constraints . . . 425
23.2.1 A Family of Self-Dual Linear Problems . . . 425
23.2.2 Embedding in a Self-Dual Problem . . . 427
23.3 Linear Problems in Standard Form . . . 429
23.3.1 The Associated Self-Dual Homogeneous System . . . 429
23.3.2 Embedding in a Feasible Self-Dual Problem . . . 430
23.4 Practical Aspects . . . 431
23.5 Extension to Linear Monotone Complementarity Problems . . . 433
23.6 Comments . . . 434

24 One-Step Methods . . . 435
24.1 Overview . . . 435
24.2 The Largest-Step Sethod . . . 436
24.2.1 Largest-Step Algorithm . . . 436
24.2.2 Largest-Step Algorithm with Safeguard . . . 436
24.3 Centralization in the Space of Large Variables . . . 437
24.3.1 One-Sided Distance . . . 437
24.3.2 Convergence with Strict Complementarity . . . 441
24.3.3 Convergence without Strict Complementarity . . . 443
24.3.4 Relative Distance in the Space of Large Variables . . . 444
24.4 Convergence Analysis . . . 445
24.4.1 Global Convergence of the Largest-Step Algorithm . . . 445
24.4.2 Local Convergence of the Largest-Step Algorithm . . . 446
24.4.3 Convergence of the Largest-Step Algorithm with Safeguard . . . 447
24.5 Comments . . . 450

25 Complexity of Linear Optimization Problems with Integer Data . . . 451
25.1 Overview . . . 451
25.2 Main Results . . . 452
25.2.1 General Hypotheses . . . 452
25.2.2 Statement of the Results . . . 452
25.2.3 Application . . . 453
25.3 Solving a System of Linear Equations . . . 453
25.4 Proofs of the Main Results . . . 455
25.4.1 Proof of Theorem 25.1 . . . 455
25.4.2 Proof of Theorem 25.2 . . . 455
25.5 Comments . . . 456

26 Karmarkar's Algorithm . . . 457
26.1 Overview . . . 457
26.2 Linear Problem in Projective Form . . . 457
26.2.1 Projective Form and Karmarkar Potential . . . 457
26.2.2 Minimizing the Potential and Solving (PLP) . . . 458
26.3 Statement of Karmarkar's Algorithm . . . 459
26.4 Analysis of the Algorithm . . . 460
26.4.1 Complexity Analysis . . . 460
26.4.2 Analysis of the Potential Decrease . . . 460
26.4.3 Estimating the Optimal Cost . . . 461
26.4.4 Practical Aspects . . . 462
26.5 Comments . . . 463

References . . . 465

Index . . . 485

Part III

Newton's Methods in Constrained Optimization

J. Charles Gilbert

In this part, we introduce and study numerical techniques based on Newton's method to solve nonlinear optimization problems: objective function and functional constraints can all be nonlinear, possibly nonconvex. Such methods, in the form called *sequential quadratic programming* (SQP), date back at least to R.B. Wilson's thesis in 1963 [359], but were mainly popularized in the mid-seventies with the appearance of their quasi-Newton versions and their globalization, see U.M. Garcia Palomares and O.L. Mangasarian [280], S.P. Han [184, 185], M.J.D. Powell [291, 292, 293], and the references therein; let us also mention the earlier contributions by B.N. Pshenichnyj [300] and S.M. Robinson [306, 307]. Ongoing research on SQP deals with the efficient use of second derivatives, particularly for nonconvex or large-scale problems, the use of trust regions [86], the treatment of singular or nearly singular situations and of equilibrium constraints [242], globalization by filters, *etc.* SQP also appears as an auxiliary tool in interior point methods for nonlinear programming [65].

Like Newton's algorithm in unconstrained optimization, SQP is more a methodology than a single algorithm. Here, the basic idea is to linearize the optimality conditions of the problem and to express the resulting linear system in a form suitable for calculation. The interest of linearization is that it provides algorithms with fast local convergence. The linear system is made up of equalities and inequalities, and is viewed as the optimality conditions of a quadratic program. Thus, SQP transforms a nonlinear optimization problem into a *sequence* of quadratic optimization problems (quadratic objective, linear equality and inequality constraints), which are simpler to solve. This process justifies the name of the SQP family of algorithms. The approach is attractive because efficient algorithms are available to solve quadratic problems: active-set methods [160, 128], augmented Lagrangian techniques [98], and interior-point methods (for the last, see part IV of the present volume).

The above-mentioned principle alone is not sufficient to derive an implementable algorithm. In fact, one must specify how to solve the quadratic program, how to deal with its possible inconsistency, how to cope with a first iterate that is far from a solution (globalization of the method), how the method can be used without computing second derivatives (quasi-Newton versions), how to take advantage of the negative curvature directions, *etc.* These questions have several answers, whose combinations result in various algorithms, more or less adapted to a particular situation. There is little to be gained from our describing each of these algorithms. Rather, our aim is to present the concepts that form the building blocks of these methods and to show why they are relevant. A good understanding of these tools should allow the reader to adapt the algorithm to a particular problem or to choose the right options of a solver, in order to make it more efficient.

The present review of Newton-like methods for constrained optimization is probably more analysis- than practice-oriented. The aim in this short account is to make an inventory of the main techniques that are continuously used to analyze these algorithms. In particular, we state and prove precise results on their properties. We also introduce and explain their structure in some detail. However, theory does not cover all aspects of an algorithm. We therefore strive to describe some heuristics that are important for efficient implementations. In fact, it is no exaggeration to say that a method is primarily judged good on the basis of its numerical efficiency. The analysis often comes afterwards to try to explain such a good behavior. Finally, let us mention that all the mathematical concepts used in the present text are simple. In particular, even though we use nonsmooth merit functions, very few notions of nonsmooth analysis are employed, so as to make the text accessible to many.

This part is organized as follows. We start in chapter 13 by recalling some theory on constrained optimization (optimality conditions, constraint qualification, projection onto a convex set, *etc.*) and Newton's method for nonlinear equations and unconstrained minimization. This chapter ends with the presentation of a numerical project that will go with us along the next chapters of this part (in §§ 14.7, 15.4, 17.4, and 18.4). This project will give us the opportunity to discuss fine points of the implementation of some of the proposed algorithms and to illustrate their behavior in various situations; it also shows, incidentally, that it is relatively easy to write one's own SQP code, provided a solver of quadratic optimization problems is available.

After these preliminaries come two chapters dealing with local methods, whose convergence is ensured if the first iterate is sufficiently close to a solution. Chapter 14 is devoted to problems with only equality constraints. Here we are in the familiar domain of Analysis, where the objects involved (functions and feasible sets) are smooth. The tools are classical as well: mainly linear algebra and differential calculus. A few concepts of differential geometry may be useful to interpret the algorithms. Chapter 15 considers the case where equalities and inequalities are present. Introducing inequality constraints results in an important additional difficulty, due to intrinsic combinatorics in the problem. This comes from the fact that one does not know a priori which inequality constraints are *active* at a solution, i.e., those that vanish at a solution. If they were known, the algorithms from chapter 14 would apply. The algorithms themselves must therefore determine the set of active constraints, among 2^{m_I} possibilities (m_I being the number of inequality constraints). Combinatorics is a serious difficulty for algorithms, but SQP copes with it by gracefully forwarding it to a quadratic subproblem, where it is easier to manage. This also implies a change of style in the analysis of the problem. Indeed, various sets of indices must be considered (active or inactive, weakly or strongly active), with an accuracy that is not obtained immediately.

The concept of exact penalty is central to force convergence of algorithms, independently of the initial iterate (a concept known as "globalization"); this is studied in chapter 16. First, the exactness properties of the Lagrangian and augmented Lagrangian can be analyzed thanks to their smoothness. These results are then used to obtain the exactness of a nondifferentiable merit function. In chapter 17, it is shown how this latter function can be used and how the local algorithms can be modified to obtain convergence of the generated iterates from a starting point that can be far from a solution. The transition from globally convergent algorithms to algorithms with rapid local convergence is also studied in that chapter.

In the quasi-Newton versions of the algorithms, the matrices containing second derivatives are replaced by matrices updated with adequate formulae; this is the subject of chapter 18.

The Problem to Solve

This text presents efficient algorithms for minimizing a real-valued function $f : \Omega \to \mathbb{R}$, defined on an *open* set Ω in $\mathbb{R}^n$, in the presence of functional constraints on the parameters $x = (x_1, \ldots, x_n)$ to optimize. Equality constraints $c_i(x) = 0$, for $i \in E$, as well as inequality constraints $c_i(x) \leq 0$, for $i \in I$, can be present. It is supposed that the index sets E (for equalities) and I (for inequalities) are *finite*, having respectively m_E and m_I elements. These constraints can also be written

$$c_E(x) = 0 \quad \text{and} \quad c_I(x) \leq 0.$$

Vector inequalities, such as $c_I(x) \leq 0$ above, are to be understood componentwise. Hence $c_I(x) \leq 0$ means that all the components of the vector $c_I(x) \in \mathbb{R}^{m_I}$ must be nonpositive. The functions f and c need not be convex.

We therefore look for a point $x_* \in \Omega$ that minimizes f on the *feasible set*

$$X = \{x \in \Omega : c_E(x) = 0,\ c_I(x) \leq 0\}.$$

A point in X is said to be *feasible*. The problem is written in a condensed way as follows:

$$(P_{EI}) \qquad \begin{cases} \min_x \ f(x) \\ c_E(x) = 0 \\ c_I(x) \leq 0 \\ x \in \Omega. \end{cases}$$

The open set Ω appearing in (P_{EI}) cannot be used to express general constraints, since a solution cannot belong to its boundary. It is simply the domain of definition of the functions f, c_E, and c_I. It is also the set where some useful properties are satisfied. For example, we always suppose that c_E is a *submersion* on Ω, i.e., that its Jacobian matrix at $x \in \Omega$,

$$A_E(x) := \nabla c_E(x)^\top,$$

of dimension $m_E \times n$ (the rows of $A_E(x)$ contain the transposed gradients $\nabla c_i(x)^\top$, $i \in E$, for the Euclidean scalar product), is surjective (or onto), for any $x \in \Omega$. Also, f and c are assumed to be smooth on Ω, for example of class C^2 (twice continuously differentiable).

We recall from definition 2.2 that problem (P_{EI}) is said to be *convex* when Ω is convex, f and the components of c_I are convex and c_E is affine. In this case, the feasible set X is convex.

Notation

We denote by

$$m = m_E + m_I$$

the total number of functional constraints. It will be often convenient to assume that E and I form a partition of $\{1, \ldots, m\}$:

$$E \cup I = \{1, \ldots, m\} \quad \text{and} \quad E \cap I = \emptyset.$$

Then, for $v \in \mathbb{R}^m$, we denote by v_E the m_E-uple made up of the components v_i of v, with indices $i \in E$; likewise for v_I. The constraints c_E and c_I are then considered to be obtained from a single function $c : \Omega \to \mathbb{R}^m$, whose components indexed in E [resp. I] form c_E [resp. c_I].

With a vector $v \in \mathbb{R}^m$, one associates the vector $v^\# \in \mathbb{R}^m$, defined as follows:

$$(v^\#)_i = \begin{cases} v_i & \text{if } i \in E \\ v_i^+ & \text{if } i \in I, \end{cases}$$

where $v_i^+ = \max(0, v_i)$. With this notation, (P_{EI}) is concisely written as:

$$\begin{cases} \min_x \ f(x) \\ c(x)^\# = 0 \\ x \in \Omega. \end{cases}$$

Indeed, $c(x)^\# = 0$ if and only if $c_E(x) = 0$ and $c_I(x) \leq 0$.

Let $x \in \Omega$. If $c_i(x) = 0$, the constraint i is said to be *active* at x. We denote by

$$I^0(x) = \{i \in I : c_i(x) = 0\}$$

the set of indices of inequality constraints that are active at $x \in \Omega$.

The Euclidean or ℓ_2 norm is denoted by $\|\cdot\|_2$. We use the same notation for the associated matrix norm.

Codes

A number of pieces of software based on the algorithmic techniques presented in this part have been written. We give a few words on some of them with a vocabulary that will be clear only after having read part III of the book.

- VF02AD by Powell [293; 1978] is part of the Harwell library. It uses Fletcher's VE02AD code (also part of the Harwell library) for solving the osculating quadratic problems [125].
- NLPQL by Schittkowski [323; 1985-86] can be found in the IMSL library. The osculating quadratic problems are solved by the dual method of Goldfarb and Idnani [165] with the modification proposed by Powell [296] (QL code).
- NPSOL by Gill, Murray, Saunders, and Wright [158; 1986] is available in the NAG library.
- FSQP by Lawrence, Tits, and Zhou [282, 222, 223, 224; 1993-2001] uses an SQP algorithm that evaluates the objective function only at points satisfying the inequality constraints. This nice property can be important for certain classes of applications.
- SPRNLP by Betts and Frank [29; 1994] can use second derivatives (if not positive definite, the Hessian of the Lagrangian is modified using a Levenberg parameter) and exploits sparsity information. It has been used to solve many optimal control problems after a direct transcription discretization.
- FAIPA by Herskovits et al. [189, 190; 1995-1998] also forces the iterates to be strictly feasible with respect to the inequality constraints. Interestingly, the algorithm requires to solve only linear systems of equations, no quadratic optimization problems [283]. This approach is connected to interior point algorithms.
- DONLP2 by Spellucci [342; 1998] is available on Netlib. It uses an active set technique on the nonlinear problem, so that the osculating quadratic problems have only equality constraints.
- SNOPT by Gill, Murray, and Saunders [156; 2002] is designed for sparse large-scale problems. The Hessian of the Lagrangian is approximated by limited memory BFGS updates (§ 6.3). The quadratic programs are solved approximately by an active set method. The globalization is done by line-search on an augmented Lagrangian merit function.
- SQPAL by Delbos, Gilbert, Glowinski, and Sinoquet [99; 2006] can solve large-scale problems since it uses an augmented Lagrangian approach for solving the quadratic problems [98], a method that has the property of identifying the active constraints in a finite number of iterations.

Notes

Surveys on Newton's method for constrained optimization have been written by Bertsekas [26; 1982], Powell [295; 1986], Fletcher [128; 1987], Gill, Murray, Saunders, and Wright [159; 1989], Spellucci [340; 1993], Boggs and Tolle [35;

1995], Polak [285; 1997], Sargent [320; 1997], Nocedal and Wright [277; 1999, Chapter 18], Conn, Gould, and Toint [86; 2000, Chapter 15], and Gould, Orban, and Toint [178; 2005]. See also [242] for problems with equilibrium constraints and [28, 325] for applications to optimal control problems.

Acknowledgement

This part of the book has benefited from the remarks and constructive comments by friends and colleagues, including Paul Armand, Laurent Chauvier, Frédéric Delbos, Sophie Jan-Jégou, Xavier Jonsson, Jean Roberts, Delphine Sinoquet, and the other authors of this book. Some students of ENSTA (École Nationale de Techniques Avancées, Paris) and DEA students of Paris I Panthéon-Sorbonne, to whom the subject of this part has been taught for several years, have also contributed to improve the accessibility of this text. Finally, I would like to thank Richard James, who kindly accepted to supervise the English phrasing of several chapters. His careful reading and erudite recommendations were greatly appreciated.

14 Local Methods for Problems with Equality Constraints

In this chapter, we present and study several local methods for minimizing a nonlinear function subject only to nonlinear equality constraints. This is the problem (P_E) represented in figure 14.1: Ω is an open set of $\mathbb{R}^n$, while

$$(P_E) \quad \begin{cases} \min_x \ f(x) \\ c(x) = 0 \\ x \in \Omega \end{cases}$$

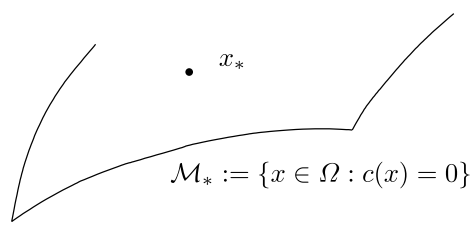

Fig. 14.1. Problem (P_E) and its feasible set

$f : \Omega \to \mathbb{R}$ and $c : \Omega \to \mathbb{R}^m$ are differentiable functions. Since we always assume that c is a submersion, which means that $c'(x)$ is surjective (or onto) for all $x \in \Omega$, the inequality $m < n$ is natural. Indeed, for the Jacobian of the constraints to be surjective, we must have $m \leq n$; but if $m = n$, any feasible point is isolated, which results in a completely different problem, for which the algorithms presented here are hardly appropriate. Therefore, a good geometrical representation of the feasible set of problem (P_E) is that of a submanifold $\mathcal{M}_*$ of $\mathbb{R}^n$, like the one depicted in figure 14.1.

There are several reasons for postponing the study of optimization problems with inequality constraints. First, we tackle difficulties and notation progressively, and prepare the intuition for the general case. Also, the reduced Hessian method (§ 14.5) has no simple equivalent form when inequalities are present. Finally, such problems arise both in their own right and as subproblems in some algorithmic approaches to solve optimization problems with inequality constraints. For instance, nonlinear interior point algorithms sometimes transform an inequality constrained problem into a sequence a equality constrained problems by introducing slack or shift variables and a logarithmic penalization (see [143, 65, 11] for examples). A good mastery of the techniques used to solve problem (P_E) is therefore helpful.

By *local methods*, we mean methods whose convergence is ensured provided the initial iterate is close enough to a solution. In this case, the algorithms presented in chapters 14 and 15 have the nice property to converge quadratically. This feature comes from the linearization of the optimality conditions. Among the quadratically convergent algorithms that have been proposed to solve problem (P_E), we have chosen to describe two of them (and some of their useful variants): Newton's method (§ 14.1) and the reduced Hessian method (§ 14.5). These are probably the most often implemented algorithms. Also, they offer a framework in which different techniques can be used: line-search and trust region globalization techniques, quasi-Newton Hessian approximations, *etc.*

When c is a submersion, the feasible set of (P_E) forms a submanifold of $\mathbb{R}^n$. However, the algorithms studied in this section do not force the iterates to stay in that manifold. For general nonlinear constraints, this would generally require too much computing time. Rather, optimality and feasibility are searched simultaneously, so that optimality is obtained in a time of the same order of magnitude as that needed to obtain feasibility in a code without optimization. This nice feature makes these algorithms very attractive in practice.

According to the first-order optimality conditions (13.1), we know that, when the constraints are qualified at a solution $x_* \in \Omega$ to (P_E), there exists a Lagrange multiplier $\lambda_* \in \mathbb{R}^m$ such that

$$\begin{cases} \nabla f(x_*) + A(x_*)^\top \lambda_* = 0 \\ c(x_*) = 0. \end{cases} \tag{14.1}$$

We have denoted by $A(x) := c'(x)$ the $m \times n$ Jacobian matrix of the constraints: the ith row of $A(x)$ is the transposed gradient $\nabla c_i(x)^\top$ of the ith constraint; hence the (i,j)th element of $A(x)$ is the partial derivative $\partial c_i / \partial x_j(x)$.

14.1 Newton's Method

The Newton Step

We have seen in chapter 13 how Newton's method can be used to solve nonlinear equations (see (13.18)) and to minimize a function (see (13.24)). For optimization problems with equality constraints, it is therefore tempting to compute the step d_k at x_k by means of a quadratic [resp. linear] approximation of the objective function [resp. constraints] at x_k. With such a method, d_k would solve or would compute a stationary point of the quadratic problem

$$\begin{cases} \min_d \ f'(x_k) \cdot d + \frac{1}{2} f''(x_k) \cdot d^2 \\ c(x_k) + c'(x_k) \cdot d = 0, \end{cases} \tag{14.2}$$

and the next iterate would be $x_{k+1} = x_k + d_k$. Beware of the **nonconvergence of this algorithm**! In some cases, the generated sequence moves away from a solution, no matter how close the initial iterate is to this solution[1].

The right approach consists in dealing simultaneously with the objective minimization and the constraint satisfaction, by working on the optimality conditions (14.1). Actually, these form a system of $n+m$ nonlinear equations in the $n+m$ unknowns (x_*, λ_*), a system that can be solved by Newton's method. This results is a so-called *primal-dual method*, which means that a sequence $\{(x_k, \lambda_k)\}$ is generated, in which x_k approximates a primal solution x_* and λ_k approximates the associated dual solution λ_*.

Let (x_k, λ_k) be the current primal-dual iterate. We use the notation

$$f_k := f(x_k), \quad c_k := c(x_k), \quad A_k := A(x_k) := c'(x_k), \quad \nabla_x \ell_k := \nabla_x \ell(x_k, \lambda_k),$$

and finally denote by

$$L_k := L(x_k, \lambda_k) := \nabla^2_{xx} \ell(x_k, \lambda_k)$$

the Hessian of the Lagrangian ℓ with respect to x at (x_k, λ_k). See (13.2) for a definition of the Lagrangian. Newton's method defines a step in (x, λ) at (x_k, λ_k) by linearizing the system (14.1) at (x_k, λ_k). One finds

$$\begin{pmatrix} L_k & A_k^\top \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ \mu_k \end{pmatrix} = - \begin{pmatrix} \nabla_x \ell_k \\ c_k \end{pmatrix}. \tag{14.3}$$

Given a solution (d_k, μ_k) to (14.3), the *Newton method* defines the next iterate (x_{k+1}, λ_{k+1}) by

$$x_{k+1} = x_k + d_k \quad \text{and} \quad \lambda_{k+1} = \lambda_k + \mu_k. \tag{14.4}$$

Since $\nabla_x \ell_k$ is linear with respect to λ_k, (14.3) can be rewritten as follows:

$$\begin{pmatrix} L_k & A_k^\top \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ \lambda_k^{\mathrm{QP}} \end{pmatrix} = - \begin{pmatrix} \nabla f_k \\ c_k \end{pmatrix}, \tag{14.5}$$

where we have used the notation

$$\lambda_k^{\mathrm{QP}} := \lambda_k + \mu_k.$$

The superscript 'QP' suggests the fact that, as we shall see below, λ_k^{QP} is the multiplier associated with the constraints of a quadratic problem. The next iterate (x_{k+1}, λ_{k+1}) of Newton's method is in this case

[1] See exercise 14.1 for an example, in which f is concave. When f is strongly convex and has a bounded Hessian, one can get convergence with line-search along the direction computed by (14.2). When f is nonconvex, convergence can still be obtained with line-search and the truncated SQP algorithm. This will be clearer with the concepts developed in chapter 17. Nevertheless, as this is shown below, the step computed by (14.2) neglects an important part of the "curvature" of problem (P_E).

$$x_{k+1} = x_k + d_k \quad \text{and} \quad \lambda_{k+1} = \lambda_k^{\mathrm{QP}}. \tag{14.6}$$

This formulation reveals the less important role played by λ_k, compared with that of x_k. Observe indeed in (14.5) that λ_k only appears in the matrix L_k, while x_k is the linearization point of the functions defining the problem.

Osculating Quadratic Problems

Just as in the unconstrained case, the Newton equation (14.3) can be viewed as the optimality system of a quadratic problem (QP), namely

$$\begin{cases} \min_d \ \nabla_x \ell_k^\top d + \frac{1}{2} d^\top L_k d \\ c_k + A_k d = 0. \end{cases} \tag{14.7}$$

This one is called the *osculating quadratic problem* of (P_E) at (x_k, λ_k). If we consider (14.5) instead of (14.3), we find

$$\begin{cases} \min_d \ \nabla f_k^\top d + \frac{1}{2} d^\top L_k d \\ c_k + A_k d = 0, \end{cases} \tag{14.8}$$

which is another osculating quadratic problem, whose optimality system is (14.5).

The transformations from (14.3) to (14.7) and from (14.5) to (14.8) call for some comments.

1. Any linear system with a symmetric matrix having the structure of that in (14.5) (the distinguishing feature is the zero $(2, 2)$ block of the matrix) can be viewed as the first order optimality conditions of the associated QP in (14.8). This point of view can be fruitful when numerical techniques to solve (14.5) are designed.
2. We know that (14.7) and (14.8) have the same primal solutions. This can also be deduced by observing that their objective functions only differ in the term $\lambda_k^\top A_k d$, which is the constant $-\lambda_k^\top c_k$ anywhere on the feasible set. However, these problems have different dual solutions. With (14.7), we obtain the step μ_k to add to the multiplier λ_k ($\lambda_{k+1} = \lambda_k + \mu_k$), while (14.8) gives directly the new multiplier ($\lambda_{k+1} = \lambda_k^{\mathrm{QP}}$).
3. One can obtain (14.7) directly from (P_E): the constraints are linearized at the current point x_k and the objective function is a quadratic approximation of the *Lagrangian* at (x_k, λ_k) (the constant term $\ell(x_k, \lambda_k)$ of this approximation can be added to the objective function of (14.7), without changing the solution).
4. Note the difference between (14.2) and (14.8). The former takes the Hessian of the objective function; the latter uses the Hessian of the Lagrangian. The difference between these two Hessians comes from the constraint curvature (sum of the terms $(\lambda_k)_i \nabla^2 c_i(x_k)$). In order to have fast

convergence, this curvature must be taken into account. This is all the more important when f is nonconvex.

The validity of (14.7) can be justified a posteriori. Indeed the Lagrangian has a minimum in the subspace tangent to the constraints (if the second-order sufficient conditions of optimality of theorem 13.4 hold); therefore, it makes sense to minimize the quadratic approximation of this Lagrangian, subject to the linearized constraints. Since the same cannot be said of f, (14.2) appears suspect.

We can also make the following remark. To have a chance of being convergent, an algorithm should at least generate a zero displacement when starting at a solution. We see that this property is not enjoyed by (14.2). In fact, if x_k solves (P_E), then $c_k = 0$ and $\nabla f(x_k)^\top d = 0$ for all $d \in N(A_k)$; hence (14.2) amounts to minimizing $\frac{1}{2} d^\top \nabla f(x_k)^2 d$ on $N(A_k)$. If the Hessian of f is not positive semi-definite in the space tangent to the constraints, which may well happen, then $d = 0$ does not solve (14.2) (unbounded problem). In contrast, (14.3) and (14.5) do enjoy this minimal property, insofar as the matrix appearing in these linear systems is nonsingular (see proposition 14.1 below and the comments that follow definition 14.2).

5. No equivalence holds between (14.5) and (14.8): the minimization problem (14.8) may have a stationary point (hence satisfying (14.5)) but no minimum (unbounded problem). Equivalence does hold between (14.5) and (14.8) – or (14.3) and (14.7) – if L_k satisfies

$$d^\top L_k d > 0, \quad \text{for all nonzero } d \text{ in } N(A_k).$$

In fact, in this case, $d \mapsto \nabla f_k^\top d + \frac{1}{2} d^\top L_k d$ is quadratic strictly convex on the affine subspace $\{d : c_k + A_k d = 0\}$. Therefore (14.8) has a unique solution, which solves the optimality equations (14.5). These equations have no other solution (proposition 14.1).

6. From a numerical point of view, the osculating quadratic problem shows that the Newton equations can be solved by minimization algorithms. For large-scale problems, the reduced conjugate gradient algorithm is often used: one computes a restoration step r_k that is feasible for (14.8) (hence satisfying $c_k + A_k r_k = 0$) and then one generates directions in the null space of A_k. We shall come back to this issue in § 14.4 and § 17.2.

Regular Stationary Points

The Newton step can be computed if the linear system that defines it, (14.5) say, is nonsingular. The next proposition gives conditions equivalent to this nonsingularity.

Proposition 14.1 (regular stationary point). *Let A be an $m \times n$ matrix, L be an $n \times n$ symmetric matrix, and*

$$K := \begin{pmatrix} L & A^\top \\ A & 0 \end{pmatrix}. \tag{14.9}$$

Then the following conditions are equivalent:

(*i*) *K is nonsingular;*

(*ii*) *A is surjective and any $d \in N(A)$ satisfying $Ld \in N(A)^\perp$ vanishes;*

(*iii*) *A is surjective and $Z^{-\top}LZ^-$ is nonsingular for some (or any) $n \times (n-m)$ matrix Z^- whose columns form a basis of $N(A)$.*

Proof. [(*i*) $\Rightarrow$ (*ii*)] Since K is surjective, so is A. On the other hand, if $d \in N(A)$ satisfies $Ld \in N(A)^\perp = R(A^\top)$, there exists $\mu \in \mathbb{R}^m$ such that $(d, \mu) \in N(K)$, so that $d = 0$.

[(*ii*) $\Rightarrow$ (*iii*)] Let Z^- be a matrix like in (*iii*). If $Z^{-\top}LZ^-u = 0$ for some $u \in \mathbb{R}^{n-m}$, $d := Z^-u \in N(A)$ and $Ld \in N(Z^{-\top}) = R(Z^-)^\perp = N(A)^\perp$, so that $Z^-u = 0$ by (*ii*). Now $u = 0$ by the injectivity of Z^-.

[(*iii*) $\Rightarrow$ (*i*)] It suffices to show that K is injective. Take (d, μ) in its null space. Then $Ad = 0$ and $Ld + A^\top\mu = 0$, which imply $d \in N(A)$ (or $d = Z^-u$ for some u) and $Z^{-\top}Ld = 0$. From (*iii*), $u = 0$ and $d = 0$. Thus $A^\top\mu = 0$, and $\mu = 0$ by the injectivity of $A^\top$. $\square$

Note that the nonsingularity of L and the surjectivity of A are not sufficient to guarantee the equivalent conditions (*i*)–(*iii*). For a counter-example consider

$$L = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{and} \qquad A = \begin{pmatrix} 1 & -1 \end{pmatrix}.$$

The vector $\begin{pmatrix} 1 & 1 & -1 \end{pmatrix}^\top$ is in the null space of K. On the other hand, when A is surjective, condition (*iii*) is obviously satisfied if $Z^{-\top}LZ^-$ is positive definite, and a fortiori if L is positive definite. Exercise 14.2 gives more information on the spectrum of the matrix K: it is claimed in particular that, when A is surjective, the matrix K always has m negative and m positive eigenvalues (for the intuition, consider the case when $n = m = 1$ and observe that the determinant of K is negative; hence there is always one negative and one positive eigenvalue).

A consequence of exercise 14.2 is that a quadratic function, whose Hessian is the matrix K with a surjective A, is never bounded below. If this function has a stationary point, it is not a minimizer, but a saddle-point. The symmetry of K suggests, however, that a linear system based on this matrix expresses the optimality conditions of a quadratic minimization problem, but this one needs linear equality constraints (using the matrix A) to have a chance of being well-posed: see (14.8) for an example. Actually, a stationary point of this constrained quadratic problem will be a constrained minimizer if and only if the matrix L is positive semi-definite on the null space of A.

The discussion above leads us to introduce the following definition.

Definition 14.2 (regular stationary point). A stationary point (x_*, λ_*) of (P_E) is said to be *regular* if $A_* := c'(x_*)$ is surjective and if $Z_*^{-\top} L_* Z_*^-$ is nonsingular, for some (or any) $n \times (n-m)$ matrix Z_*^- whose columns form a basis of $N(A_*)$.

A regular stationary point is necessarily isolated: it has a neighborhood containing no other stationary point (see exercise 14.3 for a precise statement). Also, a strong primal-dual solution (x_*, λ_*) to (P_E) satisfying (LI-CQ) (i.e., A_* surjective) is a regular stationary point. Indeed, in this case $d^\top L_* d > 0$ for all nonzero $d \in N(A_*)$, so that the so-called *reduced Hessian of the Lagrangian*

$$H_* := Z_*^{-\top} L_* Z_*^-$$

is positive definite. The $(n-m) \times (n-m)$ matrix H_* clearly depends on the choice of the matrix Z_*^-. In some cases, it can be viewed as a Hessian of some function (see exercise 14.4).

The Algorithm

We conclude this section by giving a precise description of Newton's algorithm to solve problem (P_E). As already mentioned, the method generates a primal-dual sequence $\{(x_k, \lambda_k)\} \subset \mathbb{R}^n \times \mathbb{R}^m$.

Newton's algorithm for (P_E):

Choose an initial iterate $(x_1, \lambda_1) \in \mathbb{R}^n \times \mathbb{R}^m$.
Compute $c(x_1)$, $\nabla f(x_1)$, and $A(x_1)$.
Set $k = 1$.

1. Stop if $\nabla \ell(x_k, \lambda_k) = 0$ and $c(x_k) = 0$ (optimality is reached).
2. Compute $L(x_k, \lambda_k)$ and find a primal-dual stationary point of the quadratic problem (14.8), i.e., a solution $(d_k, \lambda_k^{\mathrm{QP}})$ to (14.5).
3. Set $x_{k+1} := x_k + d_k$ and $\lambda_{k+1} := \lambda_k^{\mathrm{QP}}$.
4. Compute $c(x_{k+1})$, $\nabla f(x_{k+1})$, and $A(x_{k+1})$.
5. Increase k by 1 and go to 1.

In practice, the stopping criterion in step 1 would test whether $\|\nabla \ell(x_k, \lambda_k)\|$ and $\|c(x_k)\|$ are sufficiently small. This remark holds for all the algorithms of this part of the book.

Before analyzing the convergence properties of this algorithm in § 14.3, we introduce some notation that makes it easier to understand some interesting variants of the method and highlights the structure of the Newton step d_k. How to compute this step is dealt with in § 14.4.

14.2 Adapted Decompositions of $\mathbb{R}^n$

A General Framework

Suppose that c is a submersion on the open set $\Omega \subset \mathbb{R}^n$. Then, the set

$$\mathcal{M}_x := \{y \in \Omega : c(y) = c(x)\}$$

is a submanifold of $\mathbb{R}^n$ with dimension $n-m$ (for the few concepts of differential geometry that we use, we refer the reader to [344, 51, 84, 112] for example). Intuitively, the tangent space to $\mathcal{M}_x$ at x is the set of directions of $\mathbb{R}^n$ along which c does not vary at the first order; it is therefore the null space of the Jacobian matrix

$$A_x := A(x) := c'(x)$$

of c at x. This null space and a complementary subspace decompose $\mathbb{R}^n$ into two subspaces, which make the description and interpretation of the algorithms easier. This decomposition, which we now describe, is shown in figure 14.2.

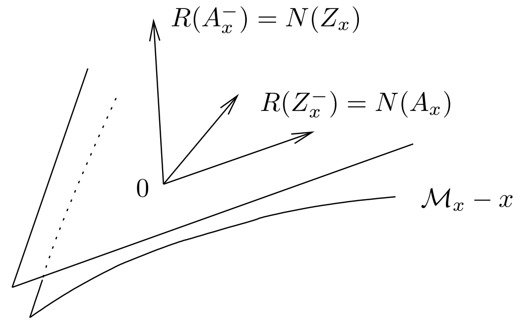

Fig. 14.2. Adapted decomposition of $\mathbb{R}^n$

Consider first the tangent subspace $N(A_x)$. We shall often assume that we have a smooth mapping

$$Z^- : \Omega \to \mathbb{R}^{n\times(n-m)} : x \mapsto Z_x^- := Z^-(x),$$

such that for all $x \in \Omega$, Z_x^- is a *basis* of the tangent subspace. We mean by this that the columns of Z_x^- form a basis of $N(A_x)$ or equivalently:

$$\forall x \in \Omega,\ Z_x^- \text{ is } n\times(n-m) \text{ injective and } A_x Z_x^- = 0. \tag{14.10}$$

Besides, since A_x is surjective, it has a *right inverse*: an $n \times m$ matrix A_x^- satisfying $A_x A_x^- = I_m$. We shall always assume that A_x^- is the value at x of a smooth mapping

$$A^- : \Omega \to \mathbb{R}^{n\times m} : x \mapsto A_x^- := A^-(x).$$

Therefore

$$\forall x \in \Omega,\ A_x^- \text{ is } n{\times}m \text{ injective and } A_x A_x^- = I_m. \tag{14.11}$$

The range space of A_x^- is a subspace complementary to $N(A_x)$, because $R(A_x^-) \cap N(A_x) = \{0\}$ and $\dim R(A_x^-) + \dim N(A_x) = m + (n{-}m) = n$.

Thus, $\mathbb{R}^n$ can be written as the direct sum of the subspaces spanned by the columns of Z_x^- and the columns of A_x^-: for all $x \in \Omega$,

$$\mathbb{R}^n = R(Z_x^-) \oplus R(A_x^-).$$

Lemma 14.3 (adapted decomposition of $\mathbb{R}^n$). *Let $Z^- : \Omega \to \mathbb{R}^{n\times(n-m)}$ and $A^- : \Omega \to \mathbb{R}^{n\times m}$ be mappings satisfying respectively* (14.10) *and* (14.11). *Then there exists a unique mapping*

$$Z : \Omega \to \mathbb{R}^{(n-m)\times n} : x \mapsto Z_x := Z(x)$$

satisfying for all $x \in \Omega$:

$$Z_x A_x^- = O_{(n-m)\times m}, \tag{14.12}$$

$$Z_x Z_x^- = I_{n-m}. \tag{14.13}$$

This mapping Z is also characterized by the following identity, valid for all $x \in \Omega$:

$$I = A_x^- A_x + Z_x^- Z_x. \tag{14.14}$$

Proof. It can be easily checked that the matrix $X_x = \begin{pmatrix} A_x^- & Z_x^- \end{pmatrix}$ is nonsingular, from which follow the existence and uniqueness of Z_x satisfying (14.12) and (14.13). Next observe from (14.10), (14.11), (14.12) and (14.13) that the matrix $Y_x = \begin{pmatrix} A_x^\top & Z_x^\top \end{pmatrix}^\top$ is the inverse of X_x, since $Y_x X_x = I_n$. Then (14.14) is exactly the identity $X_x Y_x = I_n$. Conversely, this last identity determines Y_x, hence Z_x. □

Figure 14.2 summarizes the properties of the operators A_x, Z_x^-, A_x^-, and Z_x. The manifold $\mathcal{M}_x$ is translated by $-x$, so that the linearization point x is at the origin. To find one's way in this family of operators, a mnemonic trick is welcome: the operators A_x^- and Z_x^-, with a minus exponent, are injective and right inverses; while the operators A_x and Z_x, without a minus exponent, are surjective and left inverses.

Using the identity (14.14), we have for every vector $v \in \mathbb{R}^n$,

$$v = A_x^- A_x v + Z_x^- Z_x v.$$

This identity allows us to decompose a vector v into its *longitudinal* component $Z_x^- Z_x v$, tangent at x to the manifold $\mathcal{M}_x$, and its *transversal* component

$A_x^- A_x v$, which lies in the complementary space $R(A_x^-)$. In view of our preceding development, this decomposition is well-defined, once the matrices Z_x^- and A_x^- have been given. Observe also that $A_x^- A_x$ and $Z_x^- Z_x = I - A_x^- A_x$ are oblique projectors on $R(A_x^-)$ and $R(Z_x^-)$. The orthogonal projectors on these subspaces are

$$A_x^- (A_x^{-\top} A_x^-)^{-1} A_x^{-\top} = I - Z_x^\top (Z_x Z_x^\top)^{-1} Z_x$$

and

$$Z_x^- (Z_x^{-\top} Z_x^-)^{-1} Z_x^{-\top} = I - A_x^\top (A_x A_x^\top)^{-1} A_x.$$

Below, we give some formulae for the computation of the matrices Z_x^- and A_x^- satisfying properties (14.10) and (14.11). These formulae use inverses of matrices, which need not be computed explicitly in algorithms. Likewise, the matrices Z_x^- and A_x^- need not be computed explicitly. What matters is their action (or the action of their transpose) on a vector, which can generally be obtained by solving a linear system. For example, as we shall see, the right inverse A_x^- is usually applied to the vector $c(x)$, whereas $A_x^{-\top}$ is often applied to $\nabla f(x)$.

We now proceed by giving examples of matrices Z_x^-, A_x^-, and Z_x that are frequently used in the algorithms.

Decomposition by Partitioning (or Direct Elimination)

This decomposition has its roots in *optimal control* problems (see § 1.2.2 and § 1.14 for examples of such problems), in which the variables $x = (y, u)$ are partitioned in *state variables* $y \in \mathbb{R}^m$ and *control variables* $u \in \mathbb{R}^{n-m}$. The Jacobian A_x is likewise partitioned in

$$A_x = (B_x \quad N_x),$$

where B_x is an $m \times m$ matrix giving the derivatives of the constraints with respect to the state variables. In the regular case, B_x is nonsingular. Such a decomposition is also used in linear optimization.

The decomposition of $\mathbb{R}^n$ given below is often used for large-scale optimization problems, in which a fixed partitioning of the variables leads to a nonsingular matrix B_x. Note that it is always possible to make a partition of the surjective matrix A_x as above, leading to a nonsingular matrix B_x, provided some permutation of the columns of A_x is performed. There are linear solvers that can select the columns of A_x in order to form a matrix B_x with a reasonably well optimized condition number.

In the framework just described the matrix

$$Z_x^- = \begin{pmatrix} -B_x^{-1} N_x \\ I_{n-m} \end{pmatrix} \tag{14.15}$$

is well defined and satisfies properties (14.10), while the matrix

$$A_x^- = \begin{pmatrix} B_x^{-1} \\ 0 \end{pmatrix} \tag{14.16}$$

is also well defined and satisfies (14.11). The mapping Z given by lemma 14.3 has for its value at x:

$$Z_x = \begin{pmatrix} O & I_{n-m} \end{pmatrix}.$$

Now, let us highlight some other links with the optimal control framework. Assuming that c is of class C^1, the nonsingularity of B_x implies that y, the solution to $c(y, u) = c(x)$ for fixed x, is an implicit function of u: $y = y(u)$ and $c(y(u), u) = c(x)$ for all u in a nonempty open set. Then the basis Z_x^- above is obtained by differentiating the parametrization $u \mapsto (y(u), u)$ of the manifold $\mathcal{M}_x := \{x' \in \Omega : c(x') = c(x)\}$. On the other hand, the displacement

$$-A_x^- c(x) = \begin{pmatrix} -B_x^{-1} c(x) \\ 0 \end{pmatrix}$$

is a Newton step to solve the state equation $c(y, u) = 0$, with fixed control u.

From a computational point of view, we see that, to evaluate $A_x^- c(x)$, it is sufficient to solve the linear system $B_x v = c(x)$, whose solution v gives the first m components of $A_x^- c(x)$. This is less expensive than computing B_x^{-1} explicitly! Likewise, the first m components h of $Z_x^- u$ can be obtained by solving the linear system $B_x h = -N_x u$.

Orthogonal Decomposition

The orthogonal decomposition is obtained by choosing a right inverse A_x^-, whose columns are perpendicular to $N(A_x)$ (they cannot be orthonormal in general), and a tangent basis Z_x^- with orthonormal columns. The condition on A_x^- implies that this matrix has the form $A_x^- = A_x^\top S$, for some matrix S. Since $A_x A_x^- = I$ must hold, A_x^- is necessarily given by

$$A_x^- = A_x^\top (A_x A_x^\top)^{-1}. \tag{14.17}$$

Now, let Z_x^- be an arbitrary orthonormal basis of $N(A_x)$: $A_x Z_x^- = 0$ and $Z_x^{-\top} Z_x^- = I_{n-m}$. To get the matrix Z_x provided by lemma 14.3, let us multiply both sides of the identity (14.14) to the left by $Z_x^{-\top}$, using (14.17). Necessarily

$$Z_x = Z_x^{-\top}.$$

One way of computing the matrices A_x^- and Z_x^- just described, is to use the QR factorization of $A_x^\top$ (see [170] for example):

$$A_x^\top = \begin{pmatrix} Y_x^- & Z_x^- \end{pmatrix} \begin{pmatrix} R_x \\ O \end{pmatrix} = Y_x^- R_x, \tag{14.18}$$

where $\begin{pmatrix} Y_x^- & Z_x^- \end{pmatrix}$ is an orthogonal matrix and R_x is upper triangular. The matrix R_x is nonsingular since A_x is assumed to be surjective. Then, the last $n-m$ columns Z_x^- of the orthogonal factor form an orthonormal basis of $R(Y_x^-)^\perp = R(A_x^\top)^\perp$, which is indeed the null space of A_x. Furthermore, (14.18) and the nonsingularity of R_x show that the columns of $Y_x^- \in \mathbb{R}^{n\times m}$ span $R(A_x^\top) = N(A_x)^\perp$. Since, by multiplying the extreme sides of (14.18) to the left by $Y_x^{-\top}$, it follows that $A_x Y_x^- = R_x^\top$ or $A_x Y_x^- R_x^{-\top} = I_m$, the right inverse of A_x given by (14.17) is necessarily

$$A_x^- = Y_x^- R_x^{-\top}.$$

The orthogonal decomposition just described has the advantage of being numerically stable and of computing a perfectly well-conditioned basis Z_x^-. The QR factorization can be carried out by using Givens rotations or with at most m Householder reflections. Therefore, this is a viable approach when m is not too large.

Oblique Decomposition

Let M be a matrix that is nonsingular on the null space of A_x, meaning that $Z_x^{-\top} M Z_x^-$ is nonsingular for some basis Z_x^- of $N(A_x)$ (this property of M does not depend on the choice of Z_x^-, see proposition 14.1). Then, one can associate with M a right inverse of A_x, defined as follows. Take $v \in \mathbb{R}^m$. Then the quadratic problem in d

$$\begin{cases} \min_d \ \frac{1}{2} d^\top M d \\ A_x d = v \end{cases} \tag{14.19}$$

has a unique stationary point, which satisfies the optimality conditions

$$\begin{cases} M d + A_x^\top \lambda = 0 \\ A_x d = v, \end{cases} \tag{14.20}$$

for some multiplier $\lambda \in \mathbb{R}^m$. We see that d depends linearly on v. Denoting by $\widehat{A}_x^-$ the matrix representing this linear mapping, i.e., $d = \widehat{A}_x^- v$, the second equation in (14.20) shows that $\widehat{A}_x^-$ is a right inverse of A_x. This matrix $\widehat{A}_x^-$ will be useful to write a simple expression of the Newton displacement to solve (P_E).

An explicit expression of $\widehat{A}_x^-$ can be given by using a basis Z_x^- of the null space of A_x and a right inverse A_x^- of A_x. Then (14.14) and (14.20) show that $d = A_x^- v + Z_x^- u$ for some $u \in \mathbb{R}^{n-m}$. By premultiplying both sides of the first equation of (14.20) by $Z_x^{-\top}$, we obtain $u = -(Z_x^{-\top} M Z_x^-)^{-1} Z_x^{-\top} M A_x^- v$. Finally

$$\widehat{A}_x^- = \Big(I - Z_x^- \left(Z_x^{-\top} M Z_x^- \right)^{-1} Z_x^{-\top} M \Big) A_x^-. \tag{14.21}$$

Even though A_x^- and Z_x^- appear in this formula, $\widehat{A}_x^-$ does not depend on them (from its definition). From lemma 14.3, there corresponds to the operators Z_x^-

and $\widehat{A}_x^-$ a unique matrix $\widehat{Z}_x$ such that $\widehat{Z}_x \widehat{A}_x^- = 0$ and $\widehat{Z}_x Z_x^- = I$. To give an analytic expression of $\widehat{Z}_x$, observe first that from (14.21), one has

$$Z_x^{-\top} M \widehat{A}_x^- = 0, \tag{14.22}$$

which expresses the fact that the range spaces $R(Z_x^-)$ and $R(\widehat{A}_x^-)$ are "orthogonal" with respect to the matrix M (this would correspond to a proper notion of orthogonality if the matrix M were positive definite). It is then easy to check that

$$\widehat{Z}_x = \left(Z_x^{-\top} M Z_x^-\right)^{-1} Z_x^{-\top} M$$

satisfies the required properties.

To conclude, note that $\widehat{A}_x^-$ may not exist if M is singular on the null space of A_x. Here is a counter-example with $n = 2$ and $m = 1$:

$$M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad A_x = \begin{pmatrix} 1 & 0 \end{pmatrix}, \quad \text{and} \quad Z_x^- = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Since $Z_x^{-\top} M = A_x$, $\widehat{A}_x^-$ cannot satisfy both $A_x \widehat{A}_x^- = I$ and (14.22). Observe finally that the right inverses (14.16) and (14.17) obtained previously can be recovered from $\widehat{A}_x^-$ by an appropriate choice of M; this is the subject of exercise 14.7.

14.3 Local Analysis of Newton's Method

Local Convergence

In this section, we study the local convergence of the Newton algorithm to solve problem (P_E), introduced in § 14.1. We use the notation

$$A_* = A(x_*) \quad \text{and} \quad L_* = L(x_*, \lambda_*).$$

Quadratic convergence of the primal-dual sequence $\{(x_k, \lambda_k)\}$ will be shown thanks to theorem 13.6. We shall also use proposition 14.1, whose conditions (i)-(iii) imply that the constraints are qualified at the solution x_* in the sense (LI-CQ):

$$A_* \text{ is surjective.} \tag{14.23}$$

A consequence of proposition 14.1 is that, when (x_*, λ_*) is a regular stationary point, the system (14.3) or (14.5) has a unique solution for (x_k, λ_k) close to (x_*, λ_*). Therefore Newton's method is well defined in the neighborhood of regular stationary points.

Theorem 14.4 (convergence of Newton's algorithm). *Suppose that f and c are of class C^2 in a neighborhood of a regular stationary point x_* of (P_E), with associated multiplier λ_*. Then, there exists a neighborhood V of*

(x_*, λ_*) such that, if the first iterate $(x_1, \lambda_1) \in V$, the Newton algorithm defined in § 14.1 is well-defined and generates a sequence $\{(x_k, \lambda_k)\}$ converging superlinearly to (x_*, λ_*). If f'' and c'' are Lipschitzian in a neighborhood of x_*, the convergence of the sequence is quadratic.

Proof. The result is obtained by applying theorem 13.6 with $z = (x, \lambda)$ and

$$F(z) = \begin{pmatrix} \nabla f(x) + A(x)^\top \lambda \\ c(x) \end{pmatrix}.$$

Clearly, F is of class C^1 in a neighborhood of $z_* = (x_*, \lambda_*)$ and $F'(z_*)$ is nonsingular (from proposition 14.1). The superlinear convergence of $\{(x_k, \lambda_k)\}$ to (x_*, λ_*) follows if (x_1, λ_1) is close enough to (x_*, λ_*). If f'' and c'' are Lipschitzian near x_*, so is F' near z_*, and the quadratic convergence of $\{(x_k, \lambda_k)\}$ follows. □

This theorem tells us that Newton's algorithm makes no distinction between stationary points, provided they are regular in the sense of definition 14.2. The iterates are indeed attracted by such a point, even if it is not a local minimum of (P_E); in particular it can be a maximum. The reason of this property comes from the fact that Newton's algorithm is essentially a method to solve nonlinear equations (here the optimality conditions of (P_E)). When one tries to find a minimizer, this is not a nice property. We shall see, however, that the techniques of chapter 17 tends to overcome this undesirable feature.

Note that the quadratic convergence of the sequence $\{(x_k, \lambda_k)\}$ by no means implies that of $\{x_k\}$ (see exercise 14.8). However, we shall see in chapter 15 (theorem 15.7) that $\{x_k\}$ does converge superlinearly. On the other hand, there are versions of Newton's method that guarantee the quadratic convergence of the primal sequence $\{x_k\}$. Here is an example of such an algorithm.

A Primal Version of the Newton Algorithm

It has already been observed that, in Newton's method, λ_k plays a less crucial role than x_k in the computation of the next iterate (x_{k+1}, λ_{k+1}). If, instead of letting the sequences $\{x_k\}$ and $\{\lambda_k\}$ be generated independently, the dual iterate λ_k is computed from the primal iterate x_k, by means of a function $x \mapsto \lambda(x)$, i.e.,

$$\lambda_k = \lambda(x_k),$$

the algorithm becomes completely primal. Indeed, then the knowledge of x_k entirely determines the next iterate x_{k+1}. We shall show below that the function $\lambda(\cdot)$ can be chosen in such a way that the convergence of $\{x_k\}$ will be quadratic, under natural assumptions. A possible candidate for that function is the *least-squares multiplier*:

$$\lambda^{\text{LS}}(x) := -A^{-}(x)^{\top}\nabla f(x), \tag{14.24}$$

where $A^{-}(x)$ is a right inverse of $A(x)$. One speaks of least-squares multiplier because $\lambda^{\text{LS}}(x)$ minimizes in λ a weighted ℓ_2 norm of $\nabla_x \ell(x,\lambda)$ (see exercise 14.9).

Let us make precise the algorithm under investigation.

Primal version of Newton's Algorithm for (P_E):

Choose an initial iterate $x_1 \in \mathbb{R}^n$.
Compute $c(x_1)$, $\nabla f(x_1)$, and $A(x_1)$.
Set $k = 1$.

1. Compute $\lambda_k = \lambda(x_k)$.
2. Stop if $\nabla \ell(x_k, \lambda_k) = 0$ and $c(x_k) = 0$ (optimality is reached).
3. Compute $L(x_k, \lambda_k)$ and find a solution $(d_k, \lambda_k^{\text{QP}})$ to the linear system

$$\begin{pmatrix} L(x_k,\lambda_k) & A(x_k)^{\top} \\ A(x_k) & 0 \end{pmatrix} \begin{pmatrix} d_k \\ \lambda_k^{\text{QP}} \end{pmatrix} = -\begin{pmatrix} \nabla f(x_k) \\ c(x_k) \end{pmatrix}. \tag{14.25}$$

4. Set $x_{k+1} := x_k + d_k$.
5. Compute $c(x_{k+1})$, $\nabla f(x_{k+1})$, and $A(x_{k+1})$.
6. Increase k by 1 and go to 1.

We have used the same notation λ_k^{QP} for the dual solution to (14.25) and (14.5), although their values are different, since here λ_k depends on x_k. Note that although λ_k^{QP} is computed, it has no influence on the value of x_{k+1}.

The next theorem analyses the local convergence of this algorithm.

Theorem 14.5 (convergence of a primal version of Newton's algorithm). *Suppose that f and c are of class C^2 in a neighborhood of a regular stationary point x_* of (P_E), with associated multiplier λ_*. Suppose also that the function $\lambda(\cdot)$ used to set the value of λ_k satisfies $\lambda(x_*) = \lambda_*$ and is continuous at x_*. Then, there exists a neighborhood V of x_* such that, if the first iterate $x_1 \in V$, the above primal version of Newton's algorithm is well-defined, generates a sequence $\{x_k\}$ converging superlinearly to x_*, and $\lambda_k^{\text{QP}} - \lambda_* = o(\|x_k - x_*\|)$. If furthermore f'' and c'' are Lipschitzian in a neighborhood of x_* and if there is a positive constant C such that*

$$\|\lambda(x) - \lambda_*\| \le C\|x - x_*\|, \quad \textit{for } x \textit{ near } x_*,$$

then the convergence of $\{x_k\}$ is quadratic and $\lambda_k^{\text{QP}} - \lambda_* = O(\|x_k - x_*\|^2)$.

Proof. We mimic the argument used in the proof of theorem 13.6. With the notation

$$F(x,\nu) := \begin{pmatrix} \nabla_x \ell(x,\nu) \\ c(x) \end{pmatrix},$$

and $\mu_k := \lambda_k^{\rm QP} - \lambda_*$, the linear system (14.25) can be written

$$F'(x_k,\lambda_k) \begin{pmatrix} d_k \\ \mu_k \end{pmatrix} = -F(x_k,\lambda_*).$$

If x_k is in some neighborhood of the regular stationary point x_*, with associated multiplier λ_*, λ_k is near λ_* (continuity of $\lambda(\cdot)$ at x_*). Furthermore, $F'(x_k,\lambda_k) = F'(x_k,\lambda(x_k))$ is nonsingular (see proposition 14.1) and has a bounded inverse on that neighborhood. With the notation $z_{k+1} := (x_{k+1},\lambda_k^{\rm QP})$, $z_{k,*} := (x_k,\lambda_*)$, and $z_* := (x_*,\lambda_*)$, and the fact that f and c are of class C^2, one has

$$\begin{aligned} z_{k+1} - z_* &= z_{k,*} - z_* - F'(x_k,\lambda_k)^{-1} F(x_k,\lambda_*) \\ &= F'(x_k,\lambda_k)^{-1}\Big(F'(x_k,\lambda_k)(z_{k,*}-z_*) - F(z_*) \\ &\qquad - \int_0^1 F'(x_* + t(x_k - x_*),\lambda_*)\cdot(z_{k,*}-z_*)\,\mathrm{d}t\Big). \end{aligned}$$

Using $F(z_*)=0$ and taking norms,

$$\|z_{k+1}-z_*\| \le C'\Big(\int_0^1 \|F'(x_k,\lambda_k) - F'(x_*+t(x_k-x_*),\lambda_*)\|\,\mathrm{d}t\Big)\|x_k-x_*\|,$$

where C' is a positive constant. Now, since f'', c'', and λ are continuous at x_*, $F'(\cdot,\lambda(\cdot))$ is continuous at x_* and the last estimate gives $z_{k+1}-z_* = o(\|x_k-x_*\|)$, implying the superlinear convergence of x_k to x_* and $\lambda_k^{\rm QP} - \lambda_* = o(\|x_k-x_*\|)$. If furthermore f'' and c'' are Lipschitzian near x_* and $\lambda(x)-\lambda_* = O(\|x-x_*\|)$, one has $z_{k+1}-z_* = O(\|x_k-x_*\|^2)$, which means that the convergence of $\{x_k\}$ is now quadratic and that $\lambda_k^{\rm QP}-\lambda_* = O(\|x_k-x_*\|^2)$. □

14.4 Computation of the Newton Step

In this section, we describe three ways of computing the Newton step d_k and the associated multiplier $\lambda_k^{\rm QP}$: the direct inversion approach, the dual approach, and the reduced system approach. We are interested both in analytic expressions of $(d_k,\lambda_k^{\rm QP})$ and computational issues. Each of these methods has its own advantages and drawbacks. It is the last one that most highlights the structure of the Newton step. In each case, one has to find a solution to (14.5), which is recalled here for convenience:

$$\begin{pmatrix} L_k & A_k^\top \\ A_k & 0 \end{pmatrix} \begin{pmatrix} d_k \\ \lambda_k^{\rm QP} \end{pmatrix} = -\begin{pmatrix} \nabla f_k \\ c_k \end{pmatrix}. \tag{14.26}$$

Below, the matrix of this linear system is supposed nonsingular (see proposition 14.1 for conditions ensuring this property), which implies that A_k is surjective.

The Direct Inversion Approach

The most straightforward approach for computing the Newton step is to consider the linear system (14.26) as a whole, without exploiting its block structure. One should not lose sight of the dimension $n+m$ of this linear system, which can be quite large in practice. Therefore, to make this approach attractive the problem needs to have small dimensions or to have sparse matrices L_k and A_k that can be taken into account. Using this approach could also be a naive but rapid way of computing $(d_k, \lambda_k^{\mathrm{QP}})$ in a personal program, using matrix oriented languages like Matlab or Scilab, for instance.

As regards the numerical techniques used to solve the full linear system, observe that, although the matrix in (14.26) is symmetric, it is never positive definite, even at a strong solution to problem (P_E) (see exercise 14.2). Therefore, a Cholesky factorization or conjugate gradient iterations are not adequate algorithms to solve this linear system! Direct linear solvers (i.e., those that factorize the matrix in (14.26)) can be considered, in particular when they can take advantage of the possible sparsity of A_k and L_k. The methods of Bunch and Kaufman [56] for the dense case or the MA27/MA47 solvers of Duff and Reid [114, 115, 116] for the sparse case are often employed. For large-scale problems, iterative solvers with preconditioners have also been developed, see for example [53, 20, 315, 358, 333].

The Dual Approaches

The dual approaches (sometimes called range-space approaches) need to have nonsingular matrices L_k and $A_k L_k^{-1} A_k^\top$. This condition is certainly satisfied if L_k is positive definite (remember that A_k is always assumed surjective in this section).

In the dual approach, the value of d_k is given as a function of λ_k^{QP}, using the first equation of (14.26):

$$d_k = -L_k^{-1}(\nabla f_k + A_k^\top \lambda_k^{\mathrm{QP}}). \tag{14.27}$$

Substituting this expression in the second equation of (14.26) gives the value of the QP multiplier, which is the solution to the linear system

$$(A_k L_k^{-1} A_k^\top)\lambda_k^{\mathrm{QP}} = -A_k L_k^{-1} \nabla f_k + c_k. \tag{14.28}$$

A way of solving (14.26) is then to consider the two linear systems (14.28) and (14.27) one after the other: once λ_k^{QP} has been determined by (14.28), d_k can be evaluated by (14.27). The computational effort depends on how these systems are solved, which should be a consequence of the problem size and structure. If direct solvers are used, one can give a rapid count of the number of linear systems to solve: $m{+}1$ with the $n \times n$ matrix L_k and one with the $m \times m$ matrix $A_k L_k^{-1} A_k^\top$. Indeed, the calculation can be organized as follows: first, one computes $L_k^{-1} A_k^\top$ and $L_k^{-1} \nabla f_k$; next, λ_k^{QP} is evaluated by solving (14.28); finally, d_k is obtained by (14.27) without having to solve any additional linear system.

When L_k is positive definite, λ_k^{QP} in (14.28) maximizes the dual function associated with the osculating quadratic problem (14.8), which is the function (see also part II)

$$\lambda \mapsto \min_d \left(\frac{1}{2} d^\top L_k d + \nabla f_k^\top d + \lambda^\top (c_k + A_k d) \right). \tag{14.29}$$

On the other hand, d_k given by (14.27) is the solution to this minimization problem in (14.29) with $\lambda = \lambda_k^{\text{QP}}$. This viewpoint gives its name to the approach. It also suggests other ways of solving (14.26), which are often interesting for very large-scale problems such as the Stokes equations in fluid mechanics (see [243] and references therein). We briefly discuss these approaches below.

The *Uzawa algorithm* [13, 134] generates a sequence of multipliers λ converging to λ_k^{QP}. For each λ, the minimization problem in (14.29) is solved, which provides an approximation d of the solution d_k. Next the multiplier is updated by a steepest ascent step on the dual function: $\lambda_+ := \lambda + \alpha(c_k + A_k d)$, where $\alpha > 0$ is an "appropriate" stepsize. This first order method in λ is sometimes too slow. One way of accelerating it in this simple quadratic setting is to use the conjugate gradient (CG) algorithm on the dual function, which is equivalent to solving the linear system (14.28) by CG. Each CG iteration normally requires an accurate solution to a linear system with the matrix L_k, although inexact solution can also be considered (see for example [355]).

Another way of accelerating the Uzawa procedure described above is to substitute in (14.29) the Lagrangian by the augmented Lagrangian (see § 16.3):

$$\lambda \mapsto \min_d \left(\frac{1}{2} d^\top L_k d + \nabla f_k^\top d + \lambda^\top (c_k + A_k d) + \frac{r}{2} \|c_k + A_k d\|_2^2 \right), \tag{14.30}$$

where $r > 0$ is a parameter. The algorithm is similar: $\lambda_+ := \lambda + r(c_k + A_k d)$, where d is now the solution to the minimization problem in (14.30). See [134] for more details.

Time saving is also possible by avoiding an exact minimization of the problem in (14.29) or (14.30) before updating the multiplier (see [303, 118, 54, 92] for instance).

In conclusion, the dual approaches can be appropriate when L_k and $A_k L_k^{-1} A_k^\top$ are nonsingular and a linear system with the matrix L_k is not too difficult to solve. They can also be useful when quasi-Newton techniques are used to approximate L_k^{-1} by positive definite matrices in the nonlinear algorithm (the one that sets problem (14.26)), since then there is no linear system to solve with the matrix L_k, just a matrix-vector product needs to be done.

The Reduced System Approach

In this approach (sometimes called the null-space approach), it is assumed that a decomposition of $\mathbb{R}^n$ has been chosen, similar to those described in § 14.2. The operators $A^-(x)$ and $Z^-(x)$ should take advantage of the features of the problem, in order to avoid expensive operations. We show below that then the optimization aspect contained in (14.26) can be transferred into a single linear system, involving an $(n-m) \times (n-m)$ symmetric matrix: the reduced Hessian of the Lagrangian. This makes the reduced system approach particularly appropriate when $n-m \ll n$. Since the reduced Hessian is positive definite at a strong solution to (P_E), the approach makes it possible to detect convergence to a stationary point that is not a local minimum. Furthermore, the method leads to formulae highlighting the structure of the Newton step d_k.

Let us start by introducing a very useful notion. We have denoted by $Z^-(x)$ an $n \times (n-m)$ matrix, whose columns form a basis of $N(A(x))$, the subspace tangent to the constraint manifold at x. We call *reduced gradient* of f at x for the basis Z^-, the vector of $\mathbb{R}^{n-m}$ defined by

$$g(x) := Z^-(x)^\top \nabla f(x). \tag{14.31}$$

We note $g_k := g(x_k)$. This vector can be interpreted in Riemannian geometry as follows. Equip the manifold $\mathcal{M}_x$ with a Riemannian structure by defining at each point $y \in \mathcal{M}_x$ the scalar product on the tangent space $\gamma_y(Z_y^- u, Z_y^- v) = u^\top v$; then the gradient of $f|_{\mathcal{M}_x}$ at y for this Riemannian metric is just the tangent vector $Z^-(y)g(y)$.

Consider now the computation of d_k. Recalling (14.14), the second equation in (14.26) shows that d_k has the form

$$d_k = -A_k^- c_k + Z_k^- u_k,$$

for some $u_k \in \mathbb{R}^{n-m}$. Then, the first equation in (14.26) gives

$$L_k Z_k^- u_k + A_k^\top \lambda_k^{\mathrm{QP}} = -\nabla f_k + L_k A_k^- c_k.$$

Premultiplying by $Z_k^{-\top}$ to eliminate λ_k^{QP} provides the reduced linear system:

$$H_k u_k = -g_k + Z_k^{-\top} L_k A_k^- c_k, \tag{14.32}$$

where the $(n-m) \times (n-m)$ matrix

$$H_k := Z_k^{-\top} L_k Z_k^-$$

is called the *reduced Hessian of the Lagrangian* at (x_k, λ_k). It depends on the choice of the basis Z_k^-. This matrix is necessarily nonsingular when the matrix in (14.26) is nonsingular (see proposition 14.1). This leads to

$$d_k = -(I - Z_k^- H_k^{-1} Z_k^{-\top} L_k) A_k^- c_k - Z_k^- H_k^{-1} g_k.$$

The operator acting on c_k, namely

$$\widehat{A}_k^- := (I - Z_k^- H_k^{-1} Z_k^{-\top} L_k) A_k^-, \tag{14.33}$$

is the right inverse of A_k defined in (14.21), where M and x have been replaced by L_k and x_k respectively. Finally

$$\boxed{d_k = -\widehat{A}_k^- c_k - Z_k^- H_k^{-1} g_k.} \tag{14.34}$$

This computation reveals the structure of the Newton direction d_k, made up of two terms (see figure 14.3). The first term $\widehat{r}_k := -\widehat{A}_k^- c_k$ is a stationary

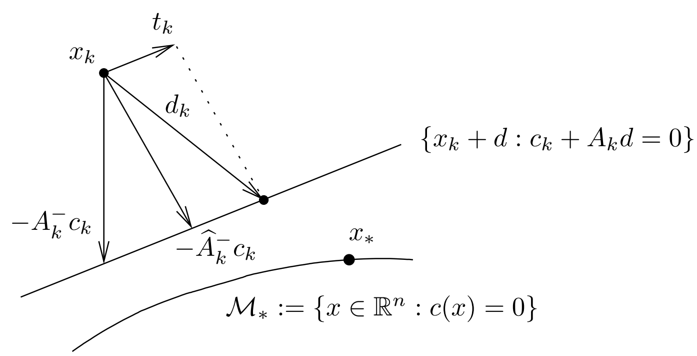

Fig. 14.3. Structure of the Newton step d_k

point of the quadratic problem in $r \in \mathbb{R}^n$:

$$\begin{cases} \min_r \ \frac{1}{2} r^\top L_k r \\ c_k + A_k r = 0. \end{cases}$$

To see this, just set $\nabla f_k = 0$ in (14.8) and (14.34). This direction aims at reducing $\rho(\cdot) = \|c(\cdot)\|$, an arbitrary norm of the constraints. Indeed, when $c_k \neq 0$, $\widehat{r}_k$ is a descent direction of ρ, since according to lemma 13.1:

$$\rho'(x_k; \widehat{r}_k) = (\|\cdot\|)'(c_k; A_k \widehat{r}_k) = (\|\cdot\|)'(c_k; -c_k) = -\|c_k\| < 0.$$

The second term in the right-hand side of (14.34), $t_k := -Z_k^- H_k^{-1} g_k$, is a stationary point of the quadratic problem in $t \in \mathbb{R}^n$:

$$\begin{cases} \min_t \ \nabla f_k^\top t + \frac{1}{2} t^\top L_k t \\ A_k t = 0. \end{cases}$$

To see this, just set $c_k = 0$ in (14.8) and (14.34). It is tangent to the manifold $\mathcal{M}_k := \mathcal{M}_{x_k}$ and aims at decreasing the function f. Indeed, when H_k is positive definite and $g_k \neq 0$, t_k is a descent direction of f at x_k, since

$$f'(x_k) \cdot t_k = \nabla f(x_k)^\top(-Z_k^- H_k^{-1} g_k) = -g_k^\top H_k^{-1} g_k < 0.$$

We shall come back to this issue in § 14.6, when comparing the direction d_k with directions generated by other algorithms.

On the influence of L_k on the direction d_k, we can observe the following.

1. The second-order information used in the Newton direction is entirely contained in the part $Z_k^{-\top} L_k$ of L_k. This can be seen in formula (14.34): only this part enters the matrices $\widehat{A}_k^-$ and H_k. In particular, the direction d_k is not changed if we add to L_k a matrix of the form $A_k^\top S_k A_k$, where S_k is an arbitrary symmetric $m \times m$ matrix.
2. If we multiply L_k by a number $\alpha \neq 0$, the transversal part $-\widehat{A}_k^- c_k$ of the direction is not affected, while the longitudinal part $-Z_k^- H_k^{-1} g_k$ is divided by α. In other words, the "size" of L_k only acts on the tangential part of d_k.

Consider now the computation of λ_k^{QP}. Premultiply the first equation of (14.26) by $A_k^{-\top}$ and use formula (14.34) of d_k to find

$$\boxed{\lambda_k^{\text{QP}} = -\widehat{A}_k^{-\top} \nabla f_k + A_k^{-\top} L_k \widehat{A}_k^- c_k.} \tag{14.35}$$

This multiplier, as well as the first term in (14.35),

$$\widehat{\lambda}_k := -\widehat{A}_k^{-\top} \nabla f_k, \tag{14.36}$$

are sometimes called *second-order multipliers*, since they involve second-order derivatives of the functions f and c, via the Hessian of the Lagrangian L_k. These are estimates of the optimal multiplier, since $\lambda_k^{\text{QP}} = \widehat{\lambda}_k = \lambda_*$ when $x_k = x_*$, a stationary point. Such is also the case of

$$\lambda_k^{\text{LS}} := -A_k^{-\top} \nabla f_k,$$

called the *first-order multiplier* or *least-squares multiplier* (see (14.24)). It is said to be of first-order because it only involves the first derivatives of the data.

With this section, we have concluded the description of Newton's algorithm to solve equality constrained optimization problems. Next comes the description of an algorithm, also proceeding by linearizations, but different from Newton's method. It can be seen as a kind of nonlinear block Gauss-Seidel approach.

14.5 Reduced Hessian Algorithm

There is an algorithm to solve problem (P_E), different from Newton's method, that also enjoys local quadratic convergence. In optimization, its existence can be suggested by the following considerations.

When c is a submersion on Ω, the feasible set

$$\mathcal{M}_* = \{x \in \Omega : c(x) = 0\}$$

is a manifold of dimension $n{-}m$. Then (P_E) has only $n{-}m$ degrees of freedom and a natural question is whether there exists a method where the matrix containing the second-order information (second derivatives of f and c or their quasi-Newton approximation) is only $(n{-}m) \times (n{-}m)$. This is certainly the case if the iterates x_k are forced to stay in $\mathcal{M}_*$. Indeed, such an algorithm can be obtained by taking a parametrization of $\mathcal{M}_*$ around x_* and applying Newton's method in the parameter space, which has dimension $n{-}m$. However, requiring $x_k \in \mathcal{M}_*$ is not realistic: it is often computationally expensive and, anyway, it cannot be realized exactly when c is an arbitrary nonlinear function. What is desired is a method with the following properties:

- the only matrix containing second-order information is $(n{-}m) \times (n{-}m)$,
- the iterates x_k are not forced to satisfy the constraints at each iteration,
- the speed of convergence is quadratic.

In this section, we show how to introduce such an algorithm. We shall see that this approach is particularly attractive when $n{-}m \ll n$ and quasi-Newton techniques are employed. Throughout the section, we assume that the stationary point x_* we are seeking is regular (see definition 14.2).

The Reduced Optimality System

The first stage leading to the definition of the algorithm is to provide an optimality system of reduced size, with fewer equations than in (14.1). This stage is optional but, by eliminating the multiplier from (14.1), it leads to a more concise presentation.

Premultiply the first equation of (14.1) by $Z^-(x_*)^\top$ to find, with (14.10), the *reduced optimality system*:

$$\begin{cases} g(x_*) = 0 \\ c(x_*) = 0, \end{cases} \tag{14.37}$$

where g is the reduced gradient of f, defined by (14.31). The multiplier λ_* no longer appears in this system, which counts $(n{-}m) + m = n$ equations for the n unknowns x_*.

Note that the two systems (14.1) and (14.37) have the same solutions x_*. Indeed, we have just shown that (14.37) can be obtained from (14.1). On the other hand, we deduce from the first equation of (14.37) that

$$\nabla f(x_*) \in N(Z^-(x_*)^\top) = R(Z^-(x_*))^\perp = N(A(x_*))^\perp = R(A(x_*)^\top).$$

Therefore there exists $\lambda_* \in \mathbb{R}^m$ such that $\nabla f(x_*) + A(x_*)^\top \lambda_* = 0$. This is the first equation of (14.1). Thus, there is no loss of solutions by considering (14.37) instead of (14.1).

Solving the Reduced Optimality System by a Decoupling Technique

The reduced Hessian method essentially consists in performing one Newton-like step to solve the second equation of (14.37), followed by one Newton-like step to solve the first equation. This resembles a nonlinear block Gauss-Seidel method. There is an important difference however. We shall show that, to yield local quadratic convergence, the first step can be an arbitrary Newton-like step, but the second one must have a very specific form. In particular, this second step must be tangent to the manifold defined by the second equation in (14.37).

The algorithm generates two sequences of iterates, $\{x_k\}$ and $\{y_k\}$, both converging to the same solution x_*. Local convergence is studied more easily if the method is thought of generating the sequence $\{y_k\}$. It is this sequence that converges (almost) quadratically. Curiously the sequence $\{x_k\}$ converges slightly less rapidly, but the algorithm is easier to implement in terms of the sequence $\{x_k\}$. We now introduce the method by considering the sequence $\{y_k\}$, while $\{x_k\}$ appears as an intermediate sequence.

Starting with an iterate $y_k \in \Omega$, we first perform a Newton-like step that aims at solving the second equation of (14.37). For this, we use a right inverse of the Jacobian of c. This gives an intermediate point x_k, defined by

$$x_k = y_k - A^-(y_k)c(y_k).$$

Note that, if $m = n$, then $A^-(y_k)$ is the inverse of $A(y_k)$ and the step $-A^-(y_k)c(y_k)$ is exactly the Newton step at y_k to solve $c(x) = 0$ (compare with (13.17) and (13.19)). When $m < n$, which is our situation, every right inverse $A^-(y_k)$ produces a particular solution $x_k - y_k$ to the constraint equation, linearized at y_k.

We are now interested in making a Newton-like step from x_k to solve the first equation of (14.37). The point x_k is supposed to be in Ω. Observe first that the reduced gradient can be written $g(x) = Z_x^{-\top}\nabla_x\ell(x,\lambda_*)$, where λ_* is the multiplier associated with the solution x_*. By optimality, $\nabla_x\ell(x_*,\lambda_*) = 0$. Hence, assuming that Z_x^- is continuous at x_* and using lemma 13.2, one has

$$g'(x_*) = Z_*^{-\top}L_*, \tag{14.38}$$

where we have set $Z_*^- = Z^-(x_*)$ and $L_* = L(x_*,\lambda_*)$, as usual. If (x_*,λ_*) is a regular stationary point, the *reduced Hessian of the Lagrangian* at (x_*,λ_*),

$$H_* := Z_*^{-\top}L_*Z_*^-,$$

is nonsingular (see proposition 14.1), so that $g'(x_*)$ is surjective. Therefore g is a submersion in a neighborhood of x_*, which is supposed to contain Ω. As above, we can therefore take a right inverse $B^-(x_k)$ of $g'(x_k)$ and define the next iterate by

$$y_{k+1} = x_k - B^-(x_k)g(x_k).$$

We have just described a procedure for computing y_{k+1} from y_k, with x_k as an intermediate iterate.

We now raise the following question. Is it possible to find a matrix mapping $x \mapsto B^-(x)$ so as to obtain fast convergence of the sequence $\{y_k\}$ to x_*? To answer this question, we introduce the functions φ and $\psi : \mathbb{R}^n \to \mathbb{R}^n$ defined by:

$$\begin{aligned}\varphi(y) &= y - A^-(y)c(y)\\ \psi(x) &= x - B^-(x)g(x).\end{aligned}$$

Then, the procedure we are analyzing can be viewed as fixed point iterations: $y_{k+1} = (\psi \circ \varphi)(y_k)$. As a result, if $B^-(\cdot)$ can be determined in such a way that $(\psi \circ \varphi)'(x_*) = 0$, the algorithm is likely to converge quadratically (see exercise 14.10). The next lemma specifies the value of $B^-_* := B^-(x_*)$ to get this property.

Lemma 14.6 (condition of quadratic convergence of a decoupling method). *Suppose that g and c are differentiable at x_*, that $A^-(\cdot)$ and $B^-(\cdot)$ are continuous at x_*, and that (x_*, λ_*) is a regular stationary point of (P_E). Then*

$$(\psi \circ \varphi)'(x_*) = 0 \quad \Longleftrightarrow \quad B^-_* = Z^-_* H_*^{-1},$$

where $H_* := Z_*^{-\top} L_* Z^-_*$, for some basis Z^-_* of $N(A_*)$.

Proof. Set $B_* = Z_*^{-\top} L_*$ and $C_* = (\psi \circ \varphi)'(x_*)$. Then, with the assumptions and lemma 13.2:

$$C_* = (I - B^-_* B_*)(I - A^-_* A_*).$$

If $(\psi \circ \varphi)'(x_*) = 0$, then $C_* Z^-_* = 0$, which gives

$$B^-_* B_* Z^-_* = Z^-_*.$$

We deduce $B^-_* = Z^-_* H_*^{-1}$. Conversely, if $B^-_* = Z^-_* H_*^{-1}$, we have $A_* B^-_* = 0$. Then

$$\begin{pmatrix} A_* \\ B_* \end{pmatrix} C_* = 0.$$

Since the operator applied to C_* is nonsingular, we have $C_* = (\psi \circ \varphi)'(x_*) = 0$. $\square$

The Algorithm

Lemma 14.6 suggests designing the algorithm that generates the sequences $\{x_k\}$ and $\{y_k\}$ as follows:

$$\begin{aligned} x_k &= y_k - A^-(y_k)c(y_k) \\ y_{k+1} &= x_k - Z^-(x_k)H_k^{-1}g(x_k). \end{aligned}$$

Here H_k is an $(n-m)\times(n-m)$ matrix approximating the reduced Hessian H_* of the Lagrangian, or $Z^-(x_k)^\top L(x_k,\lambda_k)Z^-(x_k)$, for a certain multiplier λ_k.

As such, this algorithm can be very time-consuming because the constraints must be linearized at the two points x_k and y_k, and also the two right inverses $A^-(y_k)$ and $Z^-(x_k)$ must be computed. Even though it is crucial to compute g at x_k and c at y_k, theorem 13.6 states that good convergence can be preserved if the operators involving first derivatives are evaluated at other points; the important thing is that these points converge to the solution. Since the reduced gradient must be evaluated at x_k, and since it involves a basis $Z^-(x_k)$ of the tangent space, the constraints must be linearized at x_k anyway. However, A^- can be evaluated at x_k instead of y_k. This avoids linearizing the constraints at y_k. Stating the algorithm in terms of the sequence $\{x_k\}$, we then obtain

$$\begin{aligned} y_{k+1} &= x_k - Z^-(x_k)H_k^{-1}g(x_k) \\ x_{k+1} &= y_{k+1} - A^-(x_k)c(y_{k+1}). \end{aligned}$$

Finally, setting $g_k = g(x_k)$, $A_k^- = A^-(x_k)$, $Z_k^- = Z^-(x_k)$ and

$$t_k = -Z_k^- H_k^{-1} g_k, \tag{14.39}$$

the algorithm can be stated in a very concise manner:

$$\boxed{x_{k+1} = x_k + t_k - A_k^- c(x_k+t_k).} \tag{14.40}$$

As with the Newton method (14.34), the first phase of Algorithm (14.40) consists in performing a displacement tangent to the manifold $\mathcal{M}_k$ at x_k. In the second phase, the algorithm aims at getting the next iterate x_{k+1} closer to the manifold $\mathcal{M}_*$ by taking the displacement $-A_k^- c(x_k+t_k)$, in which the constraints are evaluated at $x_k + t_k$, after the tangent step.

Although the reduced Hessian algorithm, which is summarized in the recurrence (14.40), should be quite clear, we formally state it below.

Reduced Hessian algorithm for (P_E):

Choose an initial iterate $x_1 = y_1 \in \mathbb{R}^n$.
Compute $c(x_1)$, $\nabla f(x_1)$, and $A(x_1)$.
Set $k = 1$.

1. Compute the reduced gradient $g(x_k)$ by (14.31).
2. Stop if $g(x_k) = 0$ and $c(y_k) = 0$ (optimality is reached).
3. Compute the reduced Hessian of the Lagrangian H_k, or an approximation to it, and the tangent step t_k by (14.39).

4. Evaluate the constraint at $y_{k+1} := x_k + t_k$.
5. Compute the new iterate x_{k+1} by (14.40), $\nabla f(x_{k+1})$ and $A(x_{k+1})$.
6. Increase k by 1 and go to 1.

Note that this algorithm is essentially primal, since it can be expressed only in terms of the primal sequence $\{x_k\}$. A multiplier estimate λ_k is however often necessary, either to evaluate the reduced Hessian of the Lagrangian at (x_k, λ_k) in step 3 or to update a quasi-Newton approximation to it (see chapter 18). The cheapest one is the least-squares multiplier defined by (14.24).

Simplified Newton Method

Algorithm (14.40) would be simpler if, in the second phase, the constraints were evaluated at x_k. It would then be written

$$\boxed{x_{k+1} = x_k + t_k - A_k^- c_k.} \tag{14.41}$$

This algorithm is sometimes called the *simplified Newton method* because it only uses the reduced Hessian of the Lagrangian H_k, not the full Hessian L_k (compare with (14.34) or see § 14.6). It has a slower convergence speed than (14.40): under natural assumptions, $\{x_k\}$ converges quadratically in two steps (see exercise 14.11). On the other hand, there are examples showing that the sequence $\{x_k\}$ may not converge quadratically in one step (see [63, 373]). To get good convergence, it is therefore important to evaluate the constraints at $x_k + t_k$, after the tangent displacement.

Local Convergence

The next theorem states that the sequence $\{y_k\} \equiv \{x_k + t_k\}$ of Algorithm (14.40) converges superlinearly if the matrix H_k appearing in the tangent step t_k satisfies the estimate

$$H_k - H_* = O(\|x_k - x_*\|).$$

If H_k is set to $Z^-(x_k)^\top L(x_k, \lambda_k) Z^-(x_k)$, it depends on x_k and λ_k and this condition is satisfied if $\lambda_k - \lambda_* = O(\|x_k - x_*\|)$ and if the functions f'', c'', and Z^- are Lipschitzian near x_*. This leaves a certain freedom for the choice of the multiplier λ_k. For example, one can take $\lambda_k = \lambda^{\text{LS}}(x_k)$, the least-squares multiplier defined by (14.24). It is easy to check that $\lambda^{\text{LS}}(x_*) = \lambda_*$, and thus $\lambda^{\text{LS}}(x_k) - \lambda_* = O(\|x_k - x_*\|)$ if A^- and f' are Lipschitzian near x_*. With this value of the multiplier, Algorithm (14.40) becomes entirely primal, in the sense that the algorithm only constructs the sequence $\{x_k\}$, the multiplier being reduced to an auxiliary vector, itself depending on x_k.

The result given below is slightly weaker than theorem 14.5 stating the convergence of $\{x_k\}$ in the primal variant of Newton's algorithm. For that algorithm, the sequence $\{x_k\}$ converges quadratically if $\lambda_k - \lambda_* = O(\|x_k - x_*\|)$.

The proof of theorem 14.7 uses the notation $O(\cdot)$ as explained at the end of § 13.5. At first, it may disconcert the reader. For example, the first estimate obtained in the proof, namely (14.43), means that there exists a positive constant C such that, if x_k is in some neighborhood of x_*:

$$\|y_{k+1} - x_* - (x_k - x_*) + Z_k^- H_k^{-1} Z_*^{-\top} L_*(x_k - x_*)\| \leq C\|x_k - x_*\|^2.$$

The point x_k is considered as an arbitrary point in that neighborhood and, despite the presence of the iteration index k, there is no reference to a particular sequence. The estimate obtained at the end of the proof, namely $x_{k+2} - x_* = O(\|x_k - x_*\|^2)$, implies that if x_1 and x_2 are in a sufficiently small neighborhood of x_*, then x_3 and x_4 are in that neighborhood (because for example $\|x_3 - x_*\| \leq (C\|x_1 - x_*\|)\|x_1 - x_*\| \leq \|x_1 - x_*\|$ if $\|x_1 - x_*\|$ is sufficiently small). Therefore, by induction, all the estimates can now be applied to all the generated sequences. The interest of this notation is to provide very concise proofs (for another example, see exercise 14.11).

Theorem 14.7 (convergence of the reduced Hessian algorithm). *Suppose that f and c are twice differentiable at a regular stationary point x_* of problem* (P_E) (*this allows the use of the operators $Z^-(x)$ and $A^-(x)$ introduced in* § 14.2, *for x near x_**) *and that the reduced gradient g is differentiable near x_*. Suppose also that c', g', Z^- and A^- are Lipschitzian near x_*, and that the matrix H_k used in* (14.40) *satisfies $H_k - H_* = O(\|x_k - x_*\|)$. Then, there exists a neighborhood V of x_* such that, when the first iterate $x_1 \in V$, Algorithm* (14.40) *is well defined and generates a sequence $\{x_k\}$ converging quadratically in two steps to x_*. Furthermore, the sequence $\{y_k\}$ converges superlinearly to x_* with the estimate*

$$y_{k+1} - x_* = O(\|x_{k-1} - x_*\| \, \|y_k - x_*\|). \tag{14.42}$$

Proof. Remark first that, when x_k is close to x_*, by assumption, H_k is close to H_*, which is nonsingular (x_* is regular). Thus, H_k is nonsingular and the iteration is well defined. Also $\{H_k^{-1}\}$ is bounded when x_k remains in some neighborhood of x_*.

Remembering that $y_{k+1} = x_k + t_k$ and using $g(x_*) = 0$, (14.38), and the Lipschitz continuity of g', we have

$$\begin{aligned} y_{k+1} - x_* &= x_k - x_* - Z_k^- H_k^{-1} g_k \\ &= x_k - x_* - Z_k^- H_k^{-1} Z_*^{-\top} L_*(x_k - x_*) \\ &\quad + O(\|x_k - x_*\|^2). \end{aligned} \tag{14.43}$$

But $H_k^{-1} - H_*^{-1} = -H_k^{-1}(H_k - H_*)H_*^{-1} = O(\|x_k - x_*\|)$, so that, with the Lipschitz continuity of Z^-, the following holds

$$y_{k+1} - x_* = (I - Z_*^- H_*^{-1} Z_*^{-\top} L_*)(x_k - x_*) + O(\|x_k - x_*\|^2). \tag{14.44}$$

This implies in particular that $y_{k+1}-x_* = O(\|x_k-x_*\|)$. We also have $x_{k+1} = y_{k+1} - A_k^- c(y_{k+1})$. Therefore, using successively $c(x_*) = 0$, the Lipschitz continuity of c' and A^-, (14.14), (14.44), and (14.13), we obtain

$$\begin{aligned}
x_{k+1} - x_* &= y_{k+1} - x_* - A_k^- A_*(y_{k+1} - x_*) + O(\|y_{k+1} - x_*\|^2) \\
&= y_{k+1} - x_* - A_*^- A_*(y_{k+1} - x_*) + O(\|x_k - x_*\| \, \|y_{k+1} - x_*\|) \\
&= Z_*^- Z_*(y_{k+1} - x_*) + O(\|x_k - x_*\| \, \|y_{k+1} - x_*\|) \qquad (14.45) \\
&= Z_*^- (Z_* - H_*^{-1} Z_*^{-\top} L_*)(x_k - x_*) + O(\|x_k - x_*\|^2). \qquad (14.46)
\end{aligned}$$

The operator acting on $(x_k - x_*)$ in (14.46) is nonzero in general but its square vanishes, because $(Z_* - H_*^{-1} Z_*^{-\top} L_*) Z_*^- = 0$. From this observation, we deduce the estimate

$$x_{k+2} - x_* = O(\|x_k - x_*\|^2),$$

which shows the two-step quadratic convergence of the sequence $\{x_k\}$.

Using (14.44), (14.45) (at the previous iteration), and observing that

$$(I - Z_*^- H_*^{-1} Z_*^{-\top} L_*) Z_*^- = 0,$$

we obtain (14.42). The superlinear convergence of $\{y_k\}$ follows. □

At this point it is reasonable to wonder why the convergence of the sequence $\{y_k\}$ is not quadratic. Since Algorithm (14.40) uses the second derivatives of f and c, it is legitimate to expect quadratic convergence. The above proof clarifies this, indeed: the constraints are not linearized at y_k, but at the neighboring points x_{k-1} and x_k. Then, passing from y_k to y_{k+1} involves the right inverse $A^-(x_{k-1})$ instead of $A^-(y_k)$, which perturbs the speed of convergence. If the right inverse $A^-(y_{k+1})$ were used in place of $A^-(x_k)$, an $O(\|y_{k+1} - x_*\|^2)$ would appear in (14.45) instead of an $O(\|x_k - x_*\| \, \|y_{k+1} - x_*\|)$ and quadratic convergence would ensue. Numerically, it is not clear that the computing time of $A(y_k)$ and $A^-(y_k)$ would be balanced by the quadratic convergence thus recovered, which is why the algorithm is often stated in the form (14.40).

Beware of the different behavior of the sequences $\{x_k\}$ and $\{y_k\}$. Even though they are generated by the same algorithm and both converge to the same point x_*, the first one is slower than the second one. This may look surprising, but examples do exist, in which the sequence $\{x_k\}$ does not converge quadratically (see [63]).

Newton and Quasi-Newton Versions

We have already mentioned that the reduced Hessian method is a very attractive approach when $n-m$ is much smaller than n. This is particularly true for their quasi-Newton versions. In these algorithms the $(n-m) \times (n-m)$ reduced Hessian $H_k = Z^-(x_k)^\top L(x_k, \lambda_k) Z^-(x_k)$ is approximated by a matrix updated by a quasi-Newton formula (see chapters 4.4 and 18). Only this "small" matrix needs to be updated to collect all the necessary second-order information on the problem that provides superlinear convergence. Furthermore, the small order of these updated matrices makes it possible to rapidly obtain a good approximation of the reduced Hessian.

In the Newton version, H_k must be computed. The interest of the reduced Hessian method is then less clear. One way of computing H_k is to evaluate first $L(x_k, \lambda_k)Z^-(x_k)$, by computing $n-m$ directional derivatives of the gradient of the Lagrangian along the columns of $Z^-(x_k)$, and then premultiplying the matrix thus obtained by $Z^-(x_k)^\top$. This computation is conceivable, but the knowledge of $L(x_k, \lambda_k)Z^-(x_k)$ would allow the use of Newton's method, which does not require any other information on the Hessian of the Lagrangian (see remark 1 on page 235); furthermore, Newton's method does not require a re-evaluation of the constraints after the tangent step.

Another way of getting second-order information in the reduced Hessian algorithm is to approximate H_k by computing the directional derivatives of the reduced gradient g along the $n-m$ columns of $Z^-(x_k)$. Note that $\tilde{H}_k := g'(x_k)Z^-(x_k)$ is usually different from H_k, although, in view of formula (14.38), $g'(x_*)Z_*^-$ does equal $Z_*^{-\top} L_* Z_*^-$. Now $\tilde{H}_k$ satisfies the estimate $\tilde{H}_k - H_* = O(\|x_k - x_*\|)$ (with sufficiently smooth data), so that theorem 14.7 can be applied. Note also that $\tilde{H}_k$ is not necessarily a symmetric matrix. This property depends in particular on the choice of the bases Z^-: if $Z^-(x)$ is computed by partitioning $A(x)$ (i.e., using formula (14.15)), then $\tilde{H}_k$ is symmetric; but in general it is not so when orthonormal bases are used (see [149]).

14.6 A Comparison of the Algorithms

Table 14.1 and figure 14.4 compare the form and speed of convergence of the three algorithms described in this chapter: Newton (14.6) with (14.5) or (14.34)–(14.35), simplified Newton (14.41), and reduced Hessian (14.40).

In all algorithms, the longitudinal step (tangent to the manifold $\mathcal{M}_k$) is identical and is written

$$t_k = -Z_k^- H_k^{-1} g_k.$$

When H_k is positive definite, this step is opposite to the gradient of f, seen as a function defined on the manifold $\mathcal{M}_k$ equipped at x_k with the scalar product (Riemannian structure on $\mathcal{M}_k$):

$$\gamma_{x_k}(Z_{x_k}^- u, Z_{x_k}^- v) = u^\top H_k v.$$

Algorithms	Longitudinal displacement	Transversal displacement	Speed of convergence
Newton	t_k	$-\widehat{A}_k^- c_k$	quadratic
Simplified Newton	t_k	$-A_k^- c_k$	2-step quadratic
Reduced Hessian	t_k	$-A_k^- c(x_k+t_k)$	"almost" quadratic

Table 14.1. Comparison of local methods

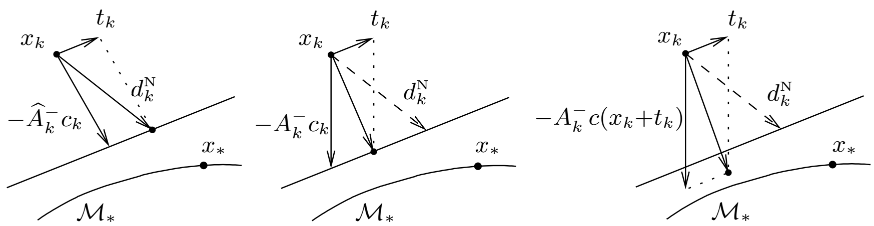

Fig. 14.4. Comparison of the Newton (d_k^{N}), simplified Newton, and reduced Hessian steps

When H_k is set to $Z_k^{-\top} L_k Z_k^-$ and H_k is positive definite, t_k can also be viewed as the unique solution to the quadratic problem in t:

$$\begin{cases} \min_t \ \nabla f_k^\top t + \frac{1}{2} t^\top L_k t \\ A_k t = 0, \end{cases}$$

This interpretation shows that t_k does not depend on the choice of the basis Z_k^-, despite the use of this matrix in the formula above. The algorithms presented in table 14.1 therefore only differ in the choice of the restoration operator, A_k^- or $\widehat{A}_k^-$, and in the points where the constraints are evaluated, x_k or x_k+t_k.

First let us compare the two forms of Newton's method: standard (step given by (14.34)), and simplified (step given by (14.41)). We see that the two displacements have the same form, but the operator acting on $c_k = c(x_k)$ is $\widehat{A}_k^-$ in the first case, and A_k^- in the second (both are right inverses of A_k). It has been observed (§ 14.2) that $\widehat{A}_k^-$ only depends on the problem's data (see problem 14.19), while A_k^- is the concern of the user of the algorithm. Theorems 14.4 and 14.5 have shown that the choice $\widehat{A}_k^-$ leads to quadratically convergent methods. On the other hand, it is easy to check that the convergence of $\{x_k\}$ with (14.41) is only two-step quadratic when the right inverse A_k^- is arbitrary: one-step quadratic convergence is never guaranteed (see exercise 14.11). Therefore Newton's method is the most effective. Note finally that one can view the simplified Newton method as an algorithm neglecting the part $Z_k^{-\top} L_k A_k^-$ of L_k in the standard Newton method (see formula (14.33)). Newton's algorithm gains in efficiency from getting more information on the Hessian of the Lagrangian.

As for the reduced Hessian algorithm (14.40), it is very close to the simplified Newton method (14.41). The algorithms differ in the point at which the constraints are evaluated: $x_k + t_k$ in (14.40) and x_k in (14.41). The reduced Hessian method can thus be viewed as a technique to compensate a possible bad choice of right inverse A_k^- by a re-evaluation of the constraints after the tangent step. As shown by theorem 14.7, this yields a good speed of convergence for the sequence $\{x_k + t_k\}$, a property that is not shared with the simplified Newton algorithm.

14.7 The Hanging Chain Project II

The goal of the second session is to implement one of the local algorithms introduced in this chapter and to understand its behavior on the hanging chain test problem presented in § 13.8 (we assume here that the main program and the simulator have been written in MATLAB). Various algorithms can be implemented. Below, we concentrate our comments on the standard Newton method described on page 221 in § 14.1, because it is this algorithm that is the easiest to extend to inequality constrained problems. We shall gain experience on its features, its efficiency, and shall reveal its weak points (some of them will be fixed in the next chapters).

We refer the reader to figure 13.3 for the general flowchart of the program. In this session, we start to write the optimization function `sqp`, which is assumed to be in the file `sqp.m`. We want to have an implementation that can be used to solve other optimization problems than the hanging chain test problem. This is a good reason for using the mathematical notation of this chapter inside `sqp.m`, not the language linked to the test problem. In our implementation, the function `sqp` has the following form

```
function [x,lme,lmi,info] = ...
    sqp (simul,x,lme,lmi,f,ce,ci,g,ae,ai,hl,options)
```

Some of the input or output arguments can be empty, depending on the presence of equality and/or inequality constraints; in particular, the variables in connection with the inequality constraints can be ignored for the while. The *input arguments* are the following: `simul` is a string giving the name of the simulator (here `'chs'`); `x` is the initial value of the primal variable x (position of the joints); `lme` and `lmi` are the initial values of the multiplier λ_E and λ_I associated with the equality and inequality constraints; `f`, `ce`, and `ci` are the values of the objective function f to minimize (the energy) and of the equality and inequality constraint functions c_E and c_I (lengths of the bars and floor constraint) at the initial point x; `g`, `ae`, and `ai` are the values of the gradient of f and the Jacobian matrices A_E and A_I of c_E and c_I at the initial point x; `hl` is the Hessian of the Lagrangian at the initial (x, λ) or an approximation to it; and the structure `options` is aimed at tuning the behavior of the solver. Standard options include upper bounds on the number of iterations and simulations (`options.iter` and `options.simul`), the required tolerances on the KKT conditions (`options.tol(1:4)`, see below), the output channel for printing (`options.fout`), *etc.* Other options will be discussed in other sessions. The *output arguments* are as follows: `x`, `lme`, and `lmi` are the final values of the primal and dual (multipliers) variables found by `sqp`; and `info` is a structure providing various information on the course of the optimization realized by the solver, telling in particular whether optimality has been reached, up to the required precision specified by the `options.tol` input argument, and in any case the reason why the solver has stopped.

We have already said on page 228 that the Newton algorithm aims at finding a stationary point, i.e., a pair (x_*, λ_*) satisfying the optimality conditions (13.1), not necessarily a local minimum. Therefore, it makes sense to have a stopping criterion based on these conditions. In our code, we stop the iterations as soon as, for some norms, the current iterate (x, λ) satisfies

$$\begin{aligned} \|\nabla f(x) + A(x)^\top \lambda\| &\leq \texttt{options.tol(1)} \\ \|c_E(x)\| &\leq \texttt{options.tol(2)} \\ \|c_I(x)^+\| &\leq \texttt{options.tol(3)} \\ \max(\|\lambda_I^-\|, \|\Lambda_I^\top c_I(x)\|) &\leq \texttt{options.tol(4)}. \end{aligned}$$

where $t^+ = \max(0, t)$, $t^- = \max(0, -t)$, and $\Lambda_I = \operatorname{Diag}(\lambda_I)$.

Writing the MATLAB function `sqp` implementing the Newton algorithm of page 221 is actually extremely simple. The core of the function is only a few lines long. The time consuming operation is the one to solve the linear system in step 2, but for a small problem this is straightforward. The easiest way of doing this operation is to form the matrix K in (14.9) and to use the standard linear solver of MATLAB (see § 14.4 for other possibilities). Since `hl` and `ae` are the variables containing respectively the Hessian of the Lagrangian and the Jacobian of the equality constraints, steps 2 and 3 of the algorithm are simply made up of

```
K = [hl ae'; ae zeros(me)];
d = -K\[g;ce];
x = x + d(1:n);
lme = d(n+1:n+me);
```

where `me` $= m_E$ is the number of equality constraints, `n` $= n$ is the number of variables, and the final values of `x` and `lme` are the updated iterates x_+ and λ_+.

Algorithmic Details, Errors to Avoid, Difficulties to Overcome

The solver `sqp` offers the user the possibility to set the initial value of x and λ. This is interesting when it is desirable to restart the solver from a known approximate solution (recall that the method is primal-dual so that both x and λ must be specified). More generally, requiring to initialize x is sensible, since the user often knows an approximate solution to the problem. This is less clear for λ, since the multipliers have sometimes a less direct "physical" meaning or, perhaps, this meaning is known but the value of λ is still difficult to determine. Therefore, it is sometimes wise to let the solver choose the initial multiplier. For an equality constrained problem, one often computes the initial λ as the solution to the linear least-squares problem

$$\min_{\lambda \in \mathbb{R}^m} \frac{1}{2} \|\nabla_x \ell(x, \lambda)\|_2^2. \tag{14.47}$$

This is motivated by the fact that the gradient of the Lagrangian vanishes at a solution. The convex quadratic problem above always has a solution (theorem 19.1), which is the least-squares multiplier (14.24) when $c'_E(x)$ is surjective.

The Newton algorithm is structured as an iteration loop, which contains the piece of code given above. Of course the simulator `simul` must be called at each iteration after having computed x_+ and λ_+, in order to update the values of `hl`, `ae`, `g`, and `ce` and to check optimality.

Writing an optimization software is a special computer science activity in the sense that the realized code has to control the convergence of a sequence. In some cases, the sequence may diverge simply because the conditions of convergence are not satisfied, not because of an error in the code. Since convergence requires an unpredictable number of iterations, it is sometimes difficult to tell on a particular case whether the behavior of the solver is correct. To certify the correctness of the function `sqp`, a good idea is to try it on problems with an increasing difficulty and to check the quadratic convergence of the generated sequences, as explained below.

- Try first to start `sqp` at the solution to a trivial problem: for example, the chain with 2 bars of length 5, with $(a, b) = (6, 0)$, whose single joint should be at position $(3, -4)$. The solver should stop without making any iteration, so that this test case checks only the validity of the stopping criterion and the simulator.
- Try next to start `sqp` near the solution to an easy problem: for example, the chain with 3 bars of length 5, with $(a, b) = (11, 0)$, whose joints should be at position $(3, -4)$ and $(8, -4)$. Convergence should be obtained in very few iterations, if the initial nodes are at positions $(2, -5)$ and $(9, -3)$. Our code converges in 5 iterations with `options.tol(1:4)` set to `1.e-10`.

The Newton algorithm of page 221 is known to converge quadratically if the initial primal-dual iterate (x_1, λ_1) is sufficiently close to a regular stationary point (theorem 14.4). Checking that quadratic convergence actually occurs is a good way of verifying that the implementation of both the algorithm and the simulator has been done properly. The very definition of quadratic convergence of a sequence $\{z_k\}$ makes use of the limit point z_* to which it converges (see § 13.5). Since, in the course of the optimization, the limit point $z_* := (x_*, \lambda_*)$ of the generated sequence $\{z_k\} := \{(x_k, \lambda_k)\}$ is not known, the definition cannot be directly applied. The idea is then to observe the behavior of another sequence, whose limit point is zero (hence known!) and that also converges quadratically. Below, we consider the following two possibilities.

- For Newton's method, a natural object to look at is the function of which the algorithm tries to find a zero. For an equality constrained optimization problem, it is the function $z := (x, \lambda) \in \mathbb{R}^{n+m} \mapsto F(z) = (\nabla_x \ell(x, \lambda), c(x)) \in \mathbb{R}^{n+m}$. When $z_* := (x_*, \lambda_*)$ is a regular stationary point (definition 14.2), $F'(z_*)$ is nonsingular and it is not difficult to show that $(z_k - z_*) \sim F(z_k)$ in the sense of (13.11). Therefore $F(z_k) \to 0$ quadratically in Newton's algorithm.
- Another vector that tends to zero is the step $s_k := z_{k+1} - z_k$. By lemma 13.5, $\{s_k\}$ also converges quadratically to zero in Newton's method.

Let us check quadratic convergence of our implementation on the following test case.

Test case 1a: second hook at $(a, b) = (1, -0.3)$, lengths of the bars: $L = (0.4,\ 0.3,\ 0.25,\ 0.2,\ 0.4)$, and initial positions of the chain joints: $(0.2, -0.5)$, $(0.4, -0.6)$, $(0.6, -0.8)$, and $(0.8, -0.6)$.

The results obtained with test case 1a are shown in figure 14.5. Convergence

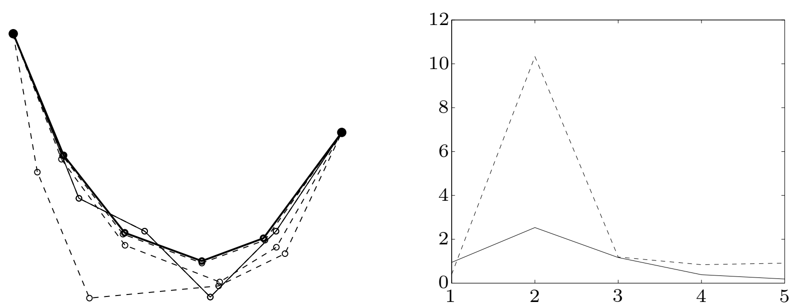

Fig. 14.5. Test case 1a

with $\texttt{options.tol}(1:4) = 10^{-10}$ is obtained in 6 iterations. The picture on the left shows the initial position of the chain (thin solid bars), the 5 intermediate positions (dashed bars) and the final position (bold solid bars). The picture on the right gives a plot of the ratios $\|F(z_{k+1})\|_2/\|F(z_k)\|_2^2$ and $\|s_{k+1}\|_2/\|s_k\|_2^2$, for $k = 1, \ldots, 5$. The boundedness of these ratios leaves no doubt on the quadratic convergence of the sequence $\{z_k\}$ to its limit.

Experimenting with the Newton Method

The test case 1a reveals the ideal behavior of Newton's method: quadratic convergence is obtained when the initial position of the chain is close to a regular solution. This solution is a strict local minimum (the smallest eigenvalue of the reduced Hessian of the Lagrangian $Z_*^{-\top} L_* Z_*^-$, for some orthonormal basis Z_*^-, is positive) and probably the global one.

Other solutions can be found by Newton's method with the same data, and those are not local minima. This is the case with the following two starting points.

Test case 1b: identical to test case 1a, except that the initial positions of the chain joints are $(0.2, 0.5)$, $(0.4, 0.6)$, $(0.6, 0.8)$, and $(0.8, 0.6)$.

Test case 1c: identical to test case 1a, except that the second hook at $(a, b) = (0.8, -0.3)$ and that the initial positions of the chain joints are $(0.3, 0.3)$, $(0.5, 0.4)$, $(0.3, 0.4)$, and $(0.6, 0.3)$.

The resulting equilibria are shown in figure 14.6. The picture on the left

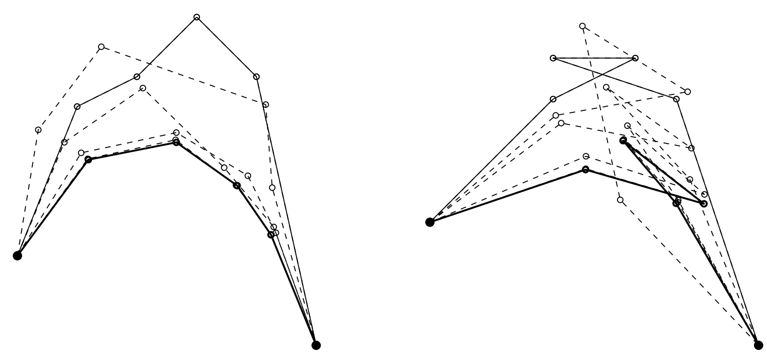

Fig. 14.6. Test cases 1b and 1c: a maximum (left) and a stationary point (right)

shows a local maximum (the largest eigenvalue of the reduced Hessian of the Lagrangian is negative). The right hand side picture shows a stationary point that is neither a minimum nor a maximum (the 3×3 reduced Hessian of the Lagrangian has two negative eigenvalues and a positive one).

The next two examples have been built to show cases without convergence.

Test case 1d: identical to test case 1a, except that the initial positions of the chain joints are $(0.2, -0.5)$, $(0.4, 1.0)$, $(0.6, -0.8)$, and $(0.8, -0.6)$ (hence, only the y-coordinate of the second joint has been modified).

Test case 2a: second hook at $(a, b) = (2, 0)$, lengths of the bars: $L = (1,\ 1)$, and initial position of the chain joint: $(1.5,\ -0.5)$.

The results are shown in figure 14.7. In the left picture, we have only plotted

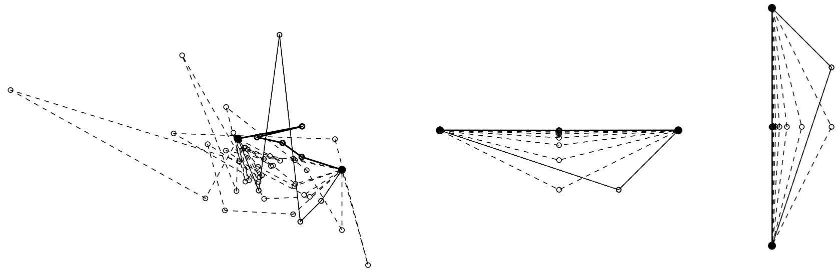

Fig. 14.7. Test cases 1d, 2a, and 2b: non convergence in (x, λ) (left), non convergence in λ (middle), and convergence in (x, λ) (right)

the position of the chain at the first 10 iterations, since apparently Newton's method does not converge. The generated sequence has a typical erratic behavior. By chance, one of these iterates may fall into the neighborhood of convergence of a stationary point, but this does not occur during the first 50 iterations. The middle picture is more puzzling, since it looks as if the algorithm converges. This is actually the case for the primal variables x (giving the position of the chain), which converge to the single feasible joint $(1, 0)$, but the dual variables diverge (their norm blows up). This reflects the fact that the optimal solution does not satisfy the KKT conditions (the Jacobian of the equality constraint in not surjective at the solution and there is no optimal multipliers); in fact, a weighty chain formed of two horizontal bars is not physically possible. The situation is quite different for the similar test case 2b below.

Test case 2b: second hook at $(a, b) = (0, -2)$, lengths of the bars: $L = (1, 1)$, and initial position of the chain joint: $(0.5, -0.5)$.

The result is shown in the right hand side picture in figure 14.7: convergence in both (x, λ) is obtained in 17 iterations.

We conclude with the following test case and let the reader guess whether the position of the chain given in figure 14.8 is a local minimum.

Test case 3: second hook at $(a, b) = (0, -1)$ and lengths of the bars: $L = (0.5, 0.5, 2.0, 0.4, 0.4)$.

Notes

The operators A^-, Z^-, and Z defined in § 14.2 were introduced by Gabay [137]. They have allowed us to use the same formalism for the optimal control and orthogonal settings. We have seen that convergence results need to have a smooth map $x \mapsto (A_x^-, Z_x^-, Z_x)$. It is usually difficult to guarantee this smoothness in a large region (for example there is no continuous basis

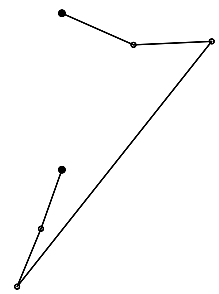

Fig. 14.8. Test case 3: is this a stable static equilibrium position?

mapping $x \mapsto Z_x^-$ on a sphere of even dimension). Even locally, standard procedures such as the QR factorization presented in § 14.2 may compute a noncontinuous basis mapping [83]. This issue has been examined by several authors, who have proposed procedures for computing a smoothly varying sequence of matrices Z_k^- when approaching a solution: see [83, 157, 24, 68]. The connection between the symmetry of $g'(x)Z^-(x)$ and the choice of basis of the tangent space is discussed in [149; § 3].

The accuracy of the computation of the Newton step by the reduced system approach (see § 14.4) crucially depends on the choice of operators A^- and Z^-. When these are obtained from the partitioning of A into $\begin{pmatrix} B & N \end{pmatrix}$, with a nonsingular B, and from a Gaussian factorization of B, Fletcher and Jonhson [129] recommend to use Gaussian elimination on the whole matrix $A^\top$ to get

$$A^\top = \begin{pmatrix} L_1 \\ L_2 \end{pmatrix} U,$$

where L_1 is unit lower triangular and U is upper triangular. The elements of L_1 and L_2 can be guaranteed to be not bigger than 1 in absolute value (e.g., because the elements of $N^\top$ are taken into account in the choice of the pivots). This approach provides well conditioned basis Z^- and a solution to the Newton system that is less sensitive to the ill-conditioning of A and that of the reduced Hessian of the Lagrangian.

The presentation of the reduced Hessian method given in § 14.5 follows [145]. This algorithm, condensed in formula (14.40), was introduced by Coleman and Conn [81], who proved convergence of the sequence $\{x_k\}$. Superlinear (or quadratic) convergence of the sequence $\{y_k\}$ was observed independently by Hoyer [197], Gilbert [145], and Byrd [64]. The simplified Newton method (14.41) has been studied by many authors: Murray and Wright [271], Powell [292], Gabay [138], Nocedal and Overton [276], Byrd and Nocedal [67], to mention a few. Newton's method on the reduced system (14.37) is considered by Goodman [177], who analyses its links with Newton's algorithm (14.5)–(14.6).

Exercises

14.1. *Nonconvergence with a step computed by* (14.2). Consider the problem in $x = (x_1, x_2) \in \mathbb{R}^2$:

$$\begin{cases} \min_x \ -ax_1^2 + 2x_2 \\ x_1^2 + x_2^2 = 1, \end{cases}$$

where $a \in \,]0,1[$. Show that the unique solution $x_* = (0, -1)$ to this problem can be repulsive for an algorithm based on (14.2): for x on the constraint manifold, arbitrary close to (but different from) the solution, and for a stationary point d of (14.2), $x + d$ is further from the solution than x.

14.2. *Inertia of the matrix K in* (14.9). The *inertia* i of a matrix is the triple (n_-, n_0, n_+) formed by the numbers of its negative, null, and positive eigenvalues respectively. Let K be the matrix defined in (14.9), where L is an $n \times n$ symmetric matrix and A is an $m \times n$ surjective matrix (hence $m \le n$). Show that

$$i(K) = i(Z^{-\top} L Z^-) + (m, 0, m),$$

where the columns of Z^- form a basis of $N(A)$ (see [90, 72, 179, 244] for related results).

[*Hint*: Prove the following claims and conclude: (*i*) $n_0(K) = n_0(Z^{-\top} L Z^-)$; (*ii*) there is no restriction in assuming that $Z^{-\top} L Z^-$ is nonsingular (use a perturbation argument, for instance), which is supposed from now on; (*iii*) $i(K) = i(Z^{-\top} L Z^-) + i(\Sigma)$, where

$$\Sigma := \begin{pmatrix} S & I_m \\ I_m & 0 \end{pmatrix}$$

for some $m \times m$ symmetric matrix S (use the matrix $\widehat{A}^-$ defined by (14.21) and Sylvester's law of inertia: $i(PKP^\top) = i(K)$ if P is nonsingular); (*iv*) $i(\Sigma) = (m, 0, m)$.]

14.3. *Regular stationary points are isolated.* Let (x_*, λ_*) be a regular stationary point of problem (P_E). Show that there is a neighborhood of (x_*, λ_*) in $\mathbb{R}^n \times \mathbb{R}^m$ containing no other stationary point than (x_*, λ_*).

14.4. *A view of the reduced Hessian of the Lagrangian.* Let $f : \Omega \to \mathbb{R}$ and $c : \Omega \to \mathbb{R}^m$ be twice differentiable functions defined in a neighborhood Ω of a point $x_* \in \mathbb{R}^n$ and denote $\ell(x, \lambda) := f(x) + \lambda^\top c(x)$, for $(x, \lambda) \in \Omega \times \mathbb{R}^m$, and $L_* := \nabla^2_{xx} \ell(x_*, \lambda_*)$. Suppose that $\nabla_x \ell(x_*, \lambda_*) = 0$ for some $\lambda_* \in \mathbb{R}^m$ (it is not assumed that $c(x_*) = 0$) and that $A_* := c'(x_*)$ is surjective. Let Z_*^- be an $n \times (n-m)$ matrix whose columns form a basis of $N(A_*)$. Show that one can find a twice differentiable parametric representation $\varphi : U \subset \mathbb{R}^{n-m} \to \mathcal{M}_{x_*} \subset \mathbb{R}^n$ of the manifold $\mathcal{M}_{x_*} := \{x \in \Omega : c(x) = c(x_*)\}$ around x_* defined in a neighborhood U of 0, such that $\varphi(0) = x_*$, $\nabla(f \circ \varphi)(0) = 0$, and $\nabla^2(f \circ \varphi)(0) = Z_*^{-\top} L_* Z_*^-$ is the reduced Hessian of the Lagrangian.

14.5. *Right inverse and complementary subspace.* Let A be an $m \times n$ surjective matrix and $\mathcal{S}$ be a subspace of $\mathbb{R}^n$, complementary to $N(A)$ (i.e., $N(A) \cap \mathcal{S} = \{0\}$ and $\dim \mathcal{S} = m$). Show that there exists a unique right inverse A^- of A such that $R(A^-) = \mathcal{S}$.

14.6. *On the orthogonal decomposition.* Let A be an $m \times n$ surjective matrix, A^- be a right inverse of A and Z^- be a matrix whose columns form a basis of $N(A)$. Show that $A^- A + Z^- Z^{-\top} = I_n$ if and only if $A^- = A^\top (AA^\top)^{-1}$ (i.e., A^- is the unique right inverse of A whose range space is perpendicular to $N(A)$) and $Z^{-\top} Z^- = I_{n-m}$ (i.e., the columns of Z^- are orthonormal).

14.7. *On the oblique right inverse.* Let A be an $m \times n$ surjective matrix. Find an $n \times n$ symmetric matrix M, that is positive definite in the null space of A, such that the right inverse $\widehat{A}^-$ of A defined by (14.20) is the one given by formula (14.16). The same question to recover the right inverse given by formula (14.17).

14.8. *Quadratic convergence of* $\{(x_k, y_k)\}$ *without linear convergence of* $\{x_k\}$. Let $y_1 \in \,]0,1[$ and consider the sequence $\{(x_k, y_k)\}_{k \geq 1} \in \mathbb{R}^2$ generated by $y_{k+1} = y_k^2$, $x_{k+1} = x_k$ if k is odd and $x_{k+1} = y_{k+1}^2$ if k is even. Show that $\{(x_k, y_k)\}$ converges quadratically to $(0,0)$, while $\{x_k\}$ does not even converge linearly to 0.

14.9. *Least-squares multiplier.* Suppose that $A(x) = c'(x)$ is surjective and let $A^-(x)$ be a right inverse of $A(x)$. Find a least-squares problem, to which the least-squares multiplier $\lambda^{\mathrm{LS}}(x) = -A^-(x)^\top \nabla f(x)$ is the solution.

[*Hint*: The *least-squares problem* has the form $\min_{\lambda \in \mathbb{R}^m} \|M \nabla_x \ell(x, \lambda)\|_2$, for some nonsingular matrix M to be found.]

14.10. *Quadratically convergent fixed point iterations.* Let $\Psi : \mathbb{R}^n \to \mathbb{R}^n$ be a $C^{1,1}$ map in the neighborhood of one of its fixed points x_* (i.e., $\Psi(x_*) = x_*$). Suppose that $\Psi'(x_*) = 0$. Show that if x_1 is sufficiently close to x_*, then the sequence generated by $x_{k+1} = \Psi(x_k)$, for $k \geq 1$, converges quadratically to x_*.

14.11. *Convergence of the simplified Newton method.* Suppose that f and c are twice differentiable at a regular stationary point x_* of problem (P_E) (this allows the use of the operators $Z^-(x)$ and $A^-(x)$ introduced in § 14.2, for x near x_*) and that the reduced gradient g is differentiable near x_*. Suppose also that c', g', Z^- and A^- are Lipschitzian near x_*, and that the matrix H_k used in the simplified Newton method (14.41) satisfies $H_k - H_* = O(\|x_k - x_*\|)$. Then, there exists a neighborhood V of x_* such that, when the first iterate $x_1 \in V$, Algorithm (14.41) is well defined and generates a sequence $\{x_k\}$ converging quadratically in two steps to x_*.

[*Hint*: Show that $x_{k+1} - x_* = Z_*^-(Z_* - H_*^{-1} Z_*^{-\top} L_*)(x_k - x_*) + O(\|x_k - x_*\|^2)$, applying a technique similar to the one used in the proof of theorem 14.7, and conclude.]

15 Local Methods for Problems with Equality and Inequality Constraints

In this chapter, we consider the general minimization problem (P_{EI}), with equality and inequality nonlinear constraints, which we recall in figure 15.1. The notation used to describe this problem was given in the introduction,

$$(P_{EI}) \qquad \begin{cases} \min_x \ f(x) \\ c_E(x) = 0 \\ c_I(x) \le 0 \\ x \in \Omega \end{cases}$$

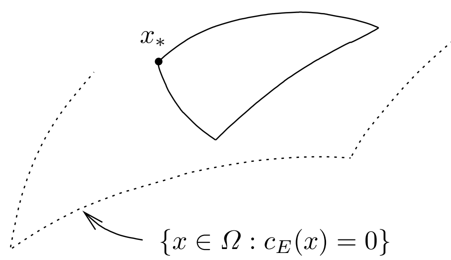

Fig. 15.1. Problem (P_{EI}) and its feasible set

on page 193. As in chapter 14, we always suppose that c_E is a submersion (i.e., $c_E'(x)$ is surjective or onto for all x in the *open* set Ω); hence the set $c_E^{-1}(0) := \{x \in \Omega : c_E(x) = 0\}$ is a submanifold of $\mathbb{R}^n$. The feasible set of (P_{EI}), denoted by

$$X := \{x \in \Omega : c_E(x) = 0,\ c_I(x) \le 0\},$$

is then the part of this manifold formed of the points also satisfying the inequality constraints $c_i(x) \le 0$ for all $i \in I$. The set delimited by the curves of $c_E^{-1}(0)$ in figure 15.1 is a typical example of feasible set for problem (P_{EI}). We have put the solution x_* on the boundary of this set, but nothing imposes that this actually occurs. The solution could just as well be inside the curved triangle without touching the solid lines. Finding a solution like the one in figure 15.1 is usually more difficult than when there is no active inequality constraints (and when this fact is known). An additional fearsome difficulty, not present in problem (P_E), is indeed linked to the determination of the active constraints at the solution.

Let us recall the first-order optimality conditions of problem (P_{EI}): when the constraints are qualified at a solution $x_* \in X$, there exists a Lagrange multiplier vector $\lambda_* \in \mathbb{R}^m$ such that

$$\text{(KKT)} \quad \begin{cases} (a)\ \nabla f(x_*) + A(x_*)^\top \lambda_* = 0 \\ (b)\ c_E(x_*) = 0, \quad c_I(x_*) \le 0 \\ (c)\ (\lambda_*)_I \ge 0 \\ (d)\ (\lambda_*)_I^\top c_I(x_*) = 0. \end{cases} \tag{15.1}$$

This chapter is organized as follows. In § 15.1, the SQP algorithm is introduced as a Newton-like approach to solve the KKT system (15.1). We shall stress the fact that, in the presence of nonconvexity, the solution to the osculating quadratic problem has to be selected with care. In § 15.2, we give conditions ensuring primal-dual quadratic convergence. First, the case when strict complementarity holds is examined. The active constraints at the solution are shown to be identified by the osculating quadratic problem as soon as the primal-dual iterate is in some neighborhood of a regular stationary point. The algorithm then reduces to Newton's method for the problem where the active constraints are considered as equality constraints, so that the local convergence result of theorem 14.4 can be applied. Next, we focus on the case without strict complementarity and show that quadratic convergence still holds, although the active constraint are no longer necessarily correctly identified by the osculating quadratic program. Necessary and sufficient conditions for primal superlinear convergence are given in § 15.3.

15.1 The SQP Algorithm

Introduction of the Algorithm

The Sequential Quadratic Programming (SQP) algorithm is a form of Newton's method to solve problem (P_{EI}) that is well adapted to computation. We have seen in chapter 14 that, to introduce such an algorithm, it is a good idea to start with the linearization of the optimality conditions and we follow the same approach here. Let us linearize (15.1) at the current point (x_k, λ_k), denoting by (d_k, μ_k) the change in the variables. This one solves the following system of equalities *and inequalities* in the unknown (d, μ):

$$\begin{cases} L_k d + A_k^\top \mu = -\nabla_x \ell_k \\ (c_k + A_k d)^{\#} = 0 \\ (\lambda_k + \mu)_I \ge 0 \\ (\lambda_k + \mu)_I^\top (c_k)_I + (\lambda_k)_I^\top (A_k d)_I = 0. \end{cases} \tag{15.2}$$

As before, we use the notation $c_k := c(x_k)$, $A_k := A(x_k) := c'(x_k)$, $\nabla_x \ell_k = \nabla_x \ell(x_k, \lambda_k)$ and $L_k := \nabla^2_{xx} \ell(x_k, \lambda_k)$. The notation $(\cdot)^{\#}$ was defined on page 194.

Because of its inequalities, (15.2) is not simple to solve. The key observation is that a good interpretation can be obtained if we add to the last equation the term $(\mu)_I^\top (A_k d)_I$. Compared with the others, this term is negligible when the steps μ_k and d_k are small, which should be the case when the iterates are close to a solution to (P_{EI}). Introducing the unknown $\lambda^{\mathrm{QP}} := \lambda_k + \mu$, the modified system (15.2) can then be written

$$\begin{cases} L_k d + A_k^\top \lambda^{\mathrm{QP}} = -\nabla f_k \\ (c_k + A_k d)^\# = 0 \\ (\lambda^{\mathrm{QP}})_I \geq 0 \\ (\lambda^{\mathrm{QP}})_I^\top (c_k + A_k d)_I = 0. \end{cases} \tag{15.3}$$

A remarkable fact, easy to check, is that (15.3) is the optimality system of the following *osculating quadratic problem* (QP)

$$\begin{cases} \min_d \nabla f(x_k)^\top d + \frac{1}{2} d^\top L_k d \\ c_E(x_k) + A_E(x_k) d = 0 \\ c_I(x_k) + A_I(x_k) d \leq 0. \end{cases} \tag{15.4}$$

This QP is easily obtained from (P_{EI}). Its constraints are those of (P_{EI}), linearized at x_k. Its objective function is hybrid, with $\nabla f(x_k)$ in the linear part and the Hessian of the Lagrangian in its quadratic part. The osculating quadratic problem (14.8), associated with the equality constrained problem (P_E), has made us familiar with the structure of (15.4).

We call *Sequential Quadratic Programming* (SQP) the algorithm generating a sequence $\{(x_k, \lambda_k)\}$ of approximations of (x_*, λ_*) by computing at each iteration a primal-dual stationary point $(d_k, \lambda_k^{\mathrm{QP}})$ of the quadratic problem (15.4), and by setting $x_{k+1} = x_k + d_k$ and $\lambda_{k+1} := \lambda_k^{\mathrm{QP}}$.

SEQUENTIAL QUADRATIC PROGRAMMING (SQP):

An initial iterate (x_1, λ_1) is given.
Compute $c(x_1)$, $\nabla f(x_1)$, and $A(x_1)$.
Set $k = 1$.

1. Stop if the KKT conditions (15.1) holds at $(x_*, \lambda_*) \equiv (x_k, \lambda_k)$ (optimality is reached).
2. Compute $L(x_k, \lambda_k)$ and find a primal-dual stationary point of (15.4), i.e., a solution $(d_k, \lambda_k^{\mathrm{QP}})$ to (15.3).
3. Set $x_{k+1} := x_k + d_k$ and $\lambda_{k+1} := \lambda_k^{\mathrm{QP}}$.
4. Compute $c(x_{k+1})$, $\nabla f(x_{k+1})$, and $A(x_{k+1})$.
5. Increase k by 1 and go to 1.

This algorithm assumes that the QP (15.4) always has a solution or, equivalently, that it is feasible and bounded (theorem 19.1). Adapted remedies must be implemented when this does not happen, such as the elastic mode of [156], which deals with infeasible linearized constraints.

What is gained with this formulation of Newton's method is that (15.4) is simpler to solve than (15.2). In fact, various quadratic programming techniques can be used to solve (15.4): active-set strategies, interior-point methods, dual approaches, *etc.* We also see that the combinatorial aspect of the original problem, which lies in the determination of the active inequality constraints, is transferred to the QP (15.4), where it is simpler to deal with than in the original nonlinear problem. However, the SQP algorithm has its own cost, which should not be overlooked. Indeed, all constraints must be linearized, including the inactive inequalities, which should play no role when the iterates are close to a solution. If these are many, the algorithm may loose some efficiency. Careful implementations use techniques to deal more efficiently with this situation (see for example [324, 301]).

Discarding Parasitic Displacements

The implementation of the SQP algorithm and the analysis of its local convergence are more complex than when only equality constraints are present. In fact, the quadratic problem (15.4) may be infeasible (its feasible set may be empty) or unbounded (the optimal value is $-\infty$), or it may have multiple local solutions (a nonconvexity effect), even in the neighborhood of a solution (x_*, λ_*) to (P_{EI}). This may happen even when (x_*, λ_*) enjoys nice properties such as the second-order sufficient conditions of optimality, strict complementarity, and constraint qualification. Here is an example.

Example 15.1. We want to minimize the logarithm of $(1+x)$ for x restricted to the interval $[0, 3]$. In canonical form, the problem is

$$\begin{cases} \min_x \, \log(1+x) \\ -x \leq 0 \\ x - 3 \leq 0. \end{cases}$$

The logarithm has been used to introduce nonconvexity in the problem, since by the monotonicity of the logarithmic function, it is equivalent to minimize $(1+x)$ or $\log(1+x)$. It is easily checked that this problem has a unique primal-dual solution $(x_*, \lambda_*) = (0, (1, 0))$, which satisfies the second-order sufficient conditions of optimality, strict complementarity, and the constraint qualification (LI-CQ). It is therefore a "good" solution. However, the osculating QP (15.4) at this solution can be written

$$\begin{cases} \min_d \, d - \frac{1}{2}d^2 \\ -d \leq 0 \\ -3 + d \leq 0. \end{cases}$$

This problem has three primal-dual stationary points (d, λ): a local minimum $(0, (1, 0))$, a maximum $(1, (0, 0))$ and a global minimum $(3, (0, 2))$. It would be unbounded without the constraint $x \leq 3$ in the original problem, which is inactive at the solution. Among these stationary points, only the first one is suitable: it gives a zero displacement (which is to be expected from an algorithm started at a solution!), and optimal multipliers. The other two stationary points are parasitic. □

The situation of this example can only occur if L_k is not positive definite. Otherwise, problem (15.4) is strictly convex and therefore has a unique solution as soon as the feasible set is nonempty. The convergence results given in § 15.2 assume that the parasitic solutions to the QP, like those revealed in the example, are discarded. Specifically, this is done by assuming that d_k is the minimum norm solution to the QP.

15.2 Primal-Dual Quadratic Convergence

We first analyze the well-posedness of the SQP algorithm and the convergence of the generated primal-dual sequences, when the first iterate is chosen in some neighborhood of a "regular" stationary point (a notion that is made precise in the statement of theorem 15.2 below) that satisfies strict complementarity. At such a stationary point, (LI-CQ) holds.

Theorem 15.2 highlights an interesting property of the SQP algorithm: in some neighborhood of a stationary point satisfying the assumptions above, the active constraints of the osculating quadratic problem (15.4) are the same as those of (P_{EI}). We have said that the identification of the active constraints is a major difficulty when solving inequality constrained problems and that, in the SQP algorithm, this difficulty is transferred to the osculating quadratic problem (QP), where it is easier to deal with. The result below tells us more: the active constraints of an osculating QP at one iteration are likely to be the same at the next iteration, at least close to a regular stationary point. Numerically, this means that, at least asymptotically, it is advantageous to solve the osculating QP's by algorithms that can take advantage of a good guess of the active constraints. Then, the combinatorial problem of determining which are the active constraints at the solution no longer occurs during the last iterations of the SQP algorithm.

Observe that, as this was already the case for equality constrained problems, the SQP algorithm may well generate a sequence that converges to a stationary point of (P_{EI}) that is not a minimum point of the problem. Observe indeed that, at any stationary point (x_*, λ_*) of (P_{EI}), $(0, \lambda_*)$ is a primal-dual solution to the quadratic problem, so that the SQP algorithm suggests not leaving x_*. This is due to the fact that SQP has been designed by linearizing the optimality conditions and therefore the algorithm makes no distinction between minima, maxima, or other stationary points.

Theorem 15.2 (primal-dual quadratic convergence of the SQP algorithm). *Suppose that f and c are of class C^2 in a neighborhood of a* *stationary point x_* of (P_{EI}), with associated multiplier λ_*. Suppose also that strict complementarity holds and that $(x_*, (\lambda_*)_{E\cup I^0_*})$ is a regular stationary point of the equality constrained problem*

$$\begin{cases} \min_x \ f(x) \\ c_i(x) = 0, \quad \text{for } i \in E \cup I^0_*, \end{cases} \tag{15.5}$$

in the sense of definition 14.2. *Consider the SQP algorithm, in which d_k is a minimum norm stationary point of the osculating quadratic problem* (15.4). *Then there is a neighborhood V of (x_*, λ_*) such that, if the first iterate $(x_1, \lambda_1) \in V$:*

- (i) *the SQP algorithm is well defined and generates a sequence $\{(x_k, \lambda_k)\}$ that converges superlinearly to (x_*, λ_*);*
- (ii) *the active constraints of the osculating quadratic problem* (15.4) *are those of problem (P_{EI});*
- (iii) *if, in addition, f and c are of class $C^{2,1}$ in a neighborhood of x_*, the convergence of $\{(x_k, \lambda_k)\}$ is quadratic.*

Proof. The idea of the proof is to show that, close to (x_*, λ_*), the selected minimum norm stationary point of the osculating quadratic problem (15.4) and the primal-dual Newton step for (15.5) are identical. The result then follows from theorem 14.4.

Suppose that (x, λ) is close to (x_*, λ_*). Since $(x_*, (\lambda_*)_{E\cup I^0_*})$ is a regular stationary point of (15.5), $c'_{E\cup I^0_*}(x_*)$ is surjective and the quadratic program in $\tilde{d}$

$$\begin{cases} \min_{\tilde{d}} \ \nabla f(x)^\top \tilde{d} + \frac{1}{2}\tilde{d}^\top L(x,\lambda)\tilde{d} \\ c_i(x) + c'_i(x)\cdot \tilde{d} = 0, \quad \text{for } i \in E \cup I^0_* \end{cases} \tag{15.6}$$

has a unique primal-dual stationary point. We denoted it by $(\tilde{d}, \tilde{\lambda}_{E\cup I^0_*})$ and form with $\tilde{\lambda}_{E\cup I^0_*}$ a vector $\tilde{\lambda} \in \mathbb{R}^m$, by setting $\tilde{\lambda}_i = 0$ for $i \in I\backslash I^0_*$.

Let us show that $(\tilde{d}, \tilde{\lambda})$ is a stationary point of the osculating quadratic problem (15.4), if $(x, \lambda) := (x_k, \lambda_k)$ is in some neighborhood of (x_*, λ_*). We only need to show that $c_i(x) + c'_i(x)\cdot \tilde{d} \le 0$ for $i \in I\backslash I^0_*$ and $\lambda_i \ge 0$ for $i \in I^0_*$. From theorem 14.4, $(x + \tilde{d}, \tilde{\lambda})$ is close to (x_*, λ_*), when (x, λ) is close to (x_*, λ_*). Therefore, for $i \in I^0_*$, $\tilde{\lambda}_i \ge 0$, since $(\lambda_*)_i > 0$ by strict complementarity. On the other hand, $\tilde{d}$ is small, so that $c_i(x) + c'_i(x)\cdot \tilde{d} \le 0$ for $i \in I\backslash I^0_*$. Hence $(\tilde{d}, \tilde{\lambda})$ is a stationary point of (15.4). We deduce from this that, for (x, λ) close to (x_*, λ_*), the SQP algorithm is well defined and d is small (it is a minimum norm stationary point and $\tilde{d}$ is small by theorem 14.4).

Let us now show that the pair $(d, \lambda^{\text{QP}}) := (d_k, \lambda_k^{\text{QP}})$ formed of the minimum norm solution to the QP and its associated multiplier is in fact $(\tilde{d}, \tilde{\lambda})$, if (x, λ) is in some neighborhood of (x_*, λ_*). From theorem 14.4, this will conclude the proof. For (x, λ) close to (x_*, λ_*) and $i \in I\backslash I^0_*$, $c_i(x) + c'_i(x)\cdot d < 0$, so that $\lambda_i^{\text{QP}} = 0 = \tilde{\lambda}_i$. Because of the uniqueness of the stationary point of (15.6), it remains to show that $c_i(x) + c'_i(x)\cdot d = 0$ for all $i \in I^0_*$ and (x, λ)

close to (x_*, λ_*). If this is not the case, there would exist an index $j \in I^0_*$ and a sequence $(x, \lambda) \to (x_*, \lambda_*)$, such that $c_j(x) + c'_j(x) \cdot d < 0$. Then $\lambda^{\rm QP}_j = 0$ and

$$\nabla f(x) + L(x, \lambda)d + \sum_{i \in (E \cup I^0_*) \setminus \{j\}} \lambda^{\rm QP}_i \nabla c_i(x) = 0.$$

Since $\nabla f(x) + L(x, \lambda)d \to \nabla f(x_*)$ (d is smaller than $\tilde{d}$, which converges to 0) and $c'_{E \cup I^0_*}(x_*)$ is surjective, $\lambda^{\rm QP}_i$ for $i \in (E \cup I^0_*) \setminus \{j\}$ would converge to some limit, $\bar{\lambda}_i$ say. Taking the limit in the equation above would give

$$\nabla f(x_*) + \sum_{i \in (E \cup I^0_*) \setminus \{j\}} \bar{\lambda}_i \nabla c_i(x_*) = 0.$$

Therefore, we would have found two different multipliers: $\bar{\lambda}$ (we set $\bar{\lambda}_i = 0$ for $i \notin (E \cup I^0_*) \setminus \{j\}$) and λ_* ($\bar{\lambda} \neq \lambda_*$ since $\bar{\lambda}_j = 0$ and $(\lambda_*)_j > 0$ by strict complementarity). This would be in contradiction with the uniqueness of the multiplier, which follows from the surjectivity of $c'_{E \cup I^0_*}(x_*)$. $\square$

It is clear from the proof of theorem 15.2 that it is not really necessary to take for d_k, a minimum norm stationary point of the osculating quadratic problem (15.4), some $d^{\min}_k$ say. The result is still true if the SQP algorithm ensures that $d_k \to 0$ when $d^{\min}_k \to 0$. For example, it would suffice to compute a stationary point d_k satisfying an estimate of the form $\|d_k\| \leq C \|d^{\min}_k\|$, for some positive constant C.

Theorem 15.4 below considers the case when strict complementarity does not hold, but assumes that (x_*, λ_*) satisfies the second order sufficient conditions of optimality and linear independence of the active constraint gradients (LI-CQ). The result is also local, in the sense that the first iterate (x_1, λ_1) is supposed to be close enough to (x_*, λ_*). The proof of this result is more difficult. This is because one can no longer use theorem 14.4 as in the preceding proof: the SQP step may be different from the Newton step on (15.5), however close to (x_*, λ_*) the current iterate (x, λ) can be. In other words, the property of local identification of the active constraints by the osculating quadratic problem no longer holds when complementarity is not strict. Here is an example.

Example 15.3. Consider the problem in $x \in \mathbb{R}$:

$$\begin{cases} \min_x \; x^2 + x^4 \\ x \leq 0. \end{cases}$$

The solution is $x_* = 0$ and $\lambda_* = 0$, so that strict complementarity does not hold. On the other hand, the constraint is qualified at x_* in the sense of (LI-CQ) and the second order sufficient conditions of optimality hold. The osculating quadratic problem at x (it does not depend on λ since the constraint is linear) is the problem in $d \in \mathbb{R}$:

$$\begin{cases} \min_d \, (2x+4x^3)d + (1+6x^2)d^2 \\ x + d \le 0. \end{cases}$$

If $x > 0$, $x + d = 0$ and the solution is obtained in one step. But if $x < 0$, $x+d = 4x^3/(1+6x^2) \in \,]2x/3, 0[$, so that the linearized constraint is inactive and the SQP step is different from the Newton step on (15.5). In this case, however, the convergence is cubic in x (also in (x, λ)): $|x + d|/|x|^3 \le 4$. $\square$

The preceding example suggests that fast convergence can still be obtained even without strict complementarity. This is confirmed by the following theorem.

Theorem 15.4 (primal-dual quadratic convergence of the SQP algorithm). *Suppose that f and c are of class $C^{2,1}$ in a neighborhood of a local solution x_* to (P_{EI}). Suppose also that the constraint qualification (LI-CQ) is satisfied at x_* and denote by λ_* the associated multiplier. Finally, suppose that the second-order sufficient condition of optimality* (13.8) *is satisfied. Consider the SQP algorithm, in which d_k is a minimum norm stationary point of the osculating quadratic problem* (15.4)*. Then there exists a neighborhood V of (x_*, λ_*) such that, if the first iterate $(x_1, \lambda_1) \in V$, the SQP algorithm is well defined and the sequence $\{(x_k, \lambda_k)\}$ converges quadratically to (x_*, λ_*).*

Proof. The following lemma is assumed (see [308]).

Lemma 15.5. *Under the conditions of theorem* 15.4*, there exists a neighborhood of (x_*, λ_*) such that* (15.3) *has a local solution and the local solution $(d_k, \lambda_k^{\rm QP})$ with d_k of minimum norm satisfies:*

$$\|d_k\| + \|\lambda_k^{\rm QP} - \lambda_*\| \le C(\|x_k - x_*\| + \|\lambda_k - \lambda_*\|).$$

From this lemma, the algorithm is well defined if (x_k, λ_k) remains close to (x_*, λ_*). This will result from the estimates obtained below.

Let us set

$$\delta_k = \|x_k - x_*\| + \|\lambda_k - \lambda_*\|.$$

From lemma 15.5, we have

$$d_k = O(\delta_k) \quad \text{and} \quad \lambda_{k+1} - \lambda_* = O(\delta_k), \tag{15.7}$$

where d_k is a minimum-norm solution to (15.4) and $\lambda_{k+1} = \lambda_k^{\rm QP}$ is the associated multiplier. We deduce that, for $i \in I \backslash I_*^0$ and δ_k small enough, we have

$$c_i(x_k) + c_i'(x_k) \cdot d_k < 0.$$

Hence $(\lambda_{k+1})_i = 0$, and with the set of indices

$$J = E \cup I_*^0,$$

the optimality of d_k is expressed by

$$L_k d_k + A_J(x_k)^\top (\lambda_{k+1})_J + \nabla f_k = 0.$$

A Taylor expansion of the left-hand side, using $\nabla_x \ell(x_*, \lambda_*) = 0$, $x_{k+1} = x_k + d_k$ and (15.7), leads to

$$\begin{aligned} 0 &= \nabla_x \ell(x_k, \lambda_*) + L(x_k, \lambda_k) d_k + A_J(x_k)^\top (\lambda_{k+1} - \lambda_*)_J \\ &= L_*(x_{k+1} - x_*) + A_J(x_*)^\top (\lambda_{k+1} - \lambda_*)_J + O(\delta_k^2). \end{aligned} \tag{15.8}$$

Expand likewise the constraints of the osculating quadratic problem: we have for $i \in J$

$$c_i(x_k) + c_i'(x_k) \cdot d_k = c_i'(x_*) \cdot (x_{k+1} - x_*) + (\gamma_k)_i, \tag{15.9}$$

where $(\gamma_k)_i = O(\delta_k^2)$.

From the assumption, $A_J(x_*)$ is surjective, so we can find a vector $v_k \in \mathbb{R}^m$ such that

$$A_J(x_*) v_k = (\gamma_k)_J \quad \text{and} \quad v_k = O(\delta_k^2).$$

The last estimate can be obtained by taking a minimum-norm v_k satisfying the first equation. With the notation

$$w_k = x_{k+1} - x_* + v_k,$$

(15.9) becomes for $i \in J$:

$$c_i(x_k) + c_i'(x_k) \cdot d_k = c_i'(x_*) \cdot w_k. \tag{15.10}$$

The complementarity conditions of the osculating quadratic problem can be written

$$(\lambda_{k+1})_i (c_i(x_k) + c_i'(x_k) \cdot d_k) = 0, \quad \text{for all } i \in I. \tag{15.11}$$

Hence, if $(\lambda_*)_i > 0$ and δ_k small enough, we have $c_i(x_k) + c_i'(x_k) \cdot d_k = 0$. Then we obtain from (15.10)

$$\begin{cases} c_i'(x_*) \cdot w_k = 0 \text{ if } i \in E \cup I_*^{0+} \\ c_i'(x_*) \cdot w_k \le 0 \text{ if } i \in I_*^{00}. \end{cases} \tag{15.12}$$

This shows that w_k lies in the critical cone C_*, defined by (13.6). From the second-order sufficiency condition, we then have for a constant $C_1 > 0$:

$$C_1 \|w_k\|^2 \le w_k^\top L_* w_k. \tag{15.13}$$

Now compute $w_k^\top L_* w_k$. From (15.8) and $v_k = O(\delta_k^2)$,

$$w_k^\top L_* w_k = -(\lambda_{k+1} - \lambda_*)_J^\top A_J(x_*) w_k + O(\|w_k\| \delta_k^2) \le C_2 \|w_k\| \delta_k^2,$$

since $(\lambda_{k+1} - \lambda_*)_J^\top A_J(x_*) w_k = 0$ thanks to (15.11) and (15.12). With (15.13), we then obtain

$$C_1\|w_k\| \le C_2\delta_k^2.$$

Since $v_k = O(\delta_k^2)$, we deduce

$$x_{k+1} - x_* = O(\delta_k^2).$$

On the other hand, this estimate, (15.8) and the injectivity of $A_J(x_*)^\top$ show that

$$(\lambda_{k+1} - \lambda_*)_J = O(\delta_k^2).$$

Since $(\lambda_{k+1})_i = (\lambda_*)_i = 0$ for $i \in I \backslash I_*^0$, these last two estimates show the quadratic convergence of the sequence $\{(x_k, \lambda_k)\}$. □

15.3 Primal Superlinear Convergence

Theorem 15.4 gives conditions for the quadratic convergence of $\{(x_k, \lambda_k)\}$. Actually, this implies neither quadratic nor superlinear convergence for $\{x_k\}$ (see exercise 14.8). Nevertheless, the following result (theorem 15.7) shows that, for the SQP algorithm using the Hessian of the Lagrangian in the quadratic programs (15.4), the sequence $\{x_k\}$ converges superlinearly. This result is interesting because it is often desirable to have fast convergence of this sequence.

We consider for this an algorithm slightly more general than the one described in § 15.1, which encompasses the quasi-Newton versions of the method. We suppose that $\{x_k\}$ is generated by

$$x_{k+1} = x_k + d_k,$$

where d_k is a stationary point of the quadratic problem

$$\begin{cases} \min_d \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ (c(x_k) + A(x_k)d)^\# = 0. \end{cases} \tag{15.14}$$

This is the same problem as (15.4), but the Hessian of the Lagrangian L_k is replaced by a symmetric matrix M_k. Incidentally, note that the multiplier λ_k is no longer explicitly used in the algorithm. Theorem 15.7 gives a necessary and sufficient condition on M_k to guarantee superlinear convergence of $\{x_k\}$.

The optimality conditions of (15.14) are (λ_k^{QP} is the multiplier associated with the constraints):

$$\begin{cases} (a)\ \nabla f_k + M_k d_k + A_k^\top \lambda_k^{\mathrm{QP}} = 0 \\ (b)\ (c_k + A_k d_k)^\# = 0 \\ (c)\ (\lambda_k^{\mathrm{QP}})_I \ge 0 \\ (d)\ (\lambda_k^{\mathrm{QP}})_I (c_k + A_k d_k)_I = 0 \end{cases} \tag{15.15}$$

We shall need the orthogonal projector onto the critical cone C_* at a solution x_* to (P_{EI}) (see (13.6)). We denote this (nonlinear) projector by P_*. It is well defined since C_* is a nonempty closed convex set.

Lemma 15.6. *If $\lambda \in \mathbb{R}^m$ is such that $\lambda_{I_*^{00}} \geq 0$ and $\lambda_{I \setminus I_*^0} = 0$, then $P_* A_*^\top \lambda = 0$.*

Proof. Take $\lambda \in \mathbb{R}^m$ as in the terms of the lemma and $h \in C_*$. Then $(A_* h)_{E \cup I_*^{0+}} = 0$, $(A_* h)_{I_*^{00}} \leq 0$, and we have

$$(0 - A_*^\top \lambda)^\top (h - 0) = -\lambda^\top A_* h = -\lambda_{I_*^{00}}^\top (A_* h)_{I_*^{00}} \geq 0.$$

The characterization (13.12) of the projection yields the result. □

Theorem 15.7 (primal superlinear convergence of the SQP algorithm). *Suppose that f and c are twice differentiable at $x_* \in \Omega$. Suppose also that (x_*, λ_*) is a primal-dual solution to (P_{EI}) satisfying (LI-CQ) and the second-order sufficient condition of optimality (13.8). Consider the sequence $\{(x_*, \lambda_*)\}$ generated by the recurrence $x_{k+1} = x_k + d_k$ and $\lambda_{k+1} = \lambda_k^{\mathrm{QP}}$, where $(d_k, \lambda_k^{\mathrm{QP}})$ is a primal-dual solution to (15.14). Suppose that $\{(x_k, \lambda_k)\}$ converges to (x_*, λ_*). Then $\{x_k\}$ converges superlinearly if and only if*

$$P_*(L_* - M_k) d_k = o(\|d_k\|), \tag{15.16}$$

where P_* is the orthogonal projector onto the critical cone C_*.

Proof. Using $(15.15)_a$, $\nabla_x \ell(x_*, \lambda_*) = 0$ and $\lambda_{k+1} \to \lambda_*$, we have

$$\begin{aligned} -M_k d_k &= \nabla_x \ell(x_k, \lambda_{k+1}) \\ &= \nabla_x \ell(x_*, \lambda_{k+1}) + L(x_*, \lambda_{k+1})(x_k - x_*) + o(\|x_k - x_*\|) \\ &= A_*^\top (\lambda_{k+1} - \lambda_*) + L_*(x_k - x_*) + o(\|x_k - x_*\|). \end{aligned}$$

Hence

$$(L_* - M_k) d_k = A_*^\top (\lambda_{k+1} - \lambda_*) + L_*(x_{k+1} - x_*) + o(\|x_k - x_*\|). \tag{15.17}$$

To show that condition (15.16) is necessary, assume that $x_{k+1} - x_* = o(\|x_k - x_*\|)$. Then (15.17) gives

$$(L_* - M_k) d_k = A_k^\top (\lambda_{k+1} - \lambda_*) + o(\|x_k - x_*\|).$$

Project with P_*, which is Lipschitzian (see (13.15)), and observe that, from $(15.15)_c$ and $(15.15)_d$, $(\lambda_{k+1} - \lambda_*)$ satisfies for large k the conditions on λ of lemma 15.6:

$$P_*(L_* - M_k) d_k = P_* A_*^\top (\lambda_{k+1} - \lambda_*) + o(\|x_k - x_*\|) = o(\|x_k - x_*\|).$$

Condition (15.16) follows, because $(x_k - x_*) \sim d_k$ by lemma 13.5.

Conversely, let us show that condition (15.16) is sufficient. For $i \in J := E \cup I_*^0$, we have

$$c_i(x_k) + c_i'(x_k) \cdot d_k = c_i'(x_*) \cdot (x_{k+1} - x_*) + (\gamma_k)_i,$$

where $(\gamma_k)_i = o(\|x_k - x_*\|) + o(\|d_k\|)$. Since $A_J(x_*)$ is surjective, $(\gamma_k)_J = A_J(x_*)v_k$, for some $v_k = o(\|x_k - x_*\|) + o(\|d_k\|)$. With the notation

$$w_k := x_{k+1} - x_* + v_k,$$

there holds

$$c_i(x_k) + c_i'(x_k) \cdot d_k = c_i'(x_*) \cdot w_k, \quad \text{for } i \in J.$$

Now $c_i(x_k) + c_i'(x_k) \cdot d_k = 0$ for $i \in E \cup I_*^{0+}$ and k large enough, so that

$$\begin{cases} c_i'(x_*) \cdot w_k = 0 \text{ if } i \in E \cup I_*^{0+} \\ c_i'(x_*) \cdot w_k \le 0 \text{ if } i \in I_*^{00}. \end{cases}$$

This implies that $w_k \in C_*$ for large k (see (13.6)) and that, for some constant $C_1 > 0$,

$$C_1 \|w_k\|^2 \le w_k^\top L_* w_k, \quad \text{for large } k. \tag{15.18}$$

On the other hand, for $i \in I_*^{00}$, from $(15.15)_d$, we have $0 = (\lambda_{k+1})_i(c_i(x_k) + c_i'(x_k) \cdot d_k) = (\lambda_{k+1})_i(c_i'(x_*) \cdot w_k)$ and $(\lambda_*)_i = 0$. While for $i \in I \backslash I_*^0$, $(\lambda_{k+1} - \lambda_*)_i = 0$. Therefore

$$(\lambda_{k+1} - \lambda_*)^\top A_* w_k = 0, \quad \text{for large } k.$$

Now, with this equation, (15.17), $v_k = o(\|x_k - x_*\|) + o(\|d_k\|)$, the fact that $u^\top v \le u^\top P_* v$, for all $v \in \mathbb{R}^n$ and all $u \in C_*$ (see (13.14)), and (15.16), we find that

$$\begin{aligned} w_k^\top L_* w_k &= w_k^\top L_*(x_{k+1} - x_*) + O(\|w_k\| \, \|v_k\|) \\ &= w_k^\top (L_* - M_k) d_k + o(\|w_k\| \, \|x_k - x_*\|) + o(\|w_k\| \, \|d_k\|) \\ &\le w_k^\top P_*(L_* - M_k) d_k + o(\|w_k\| \, \|x_k - x_*\|) + o(\|w_k\| \, \|d_k\|) \\ &= o(\|w_k\| \, \|x_k - x_*\|) + o(\|w_k\| \, \|d_k\|). \end{aligned}$$

With (15.18), $w_k = o(\|x_k - x_*\|) + o(\|d_k\|)$; hence

$$x_{k+1} - x_* = o(\|x_k - x_*\|) + o(\|d_k\|).$$

The property $x_{k+1} - x_* = o(\|x_k - x_*\|)$ follows easily. □

When there are no inequality constraints, P_* is the orthogonal projector onto the null space $N(A_*)$. It is then linear. Given a basis Z_*^- of $N(A_*)$, it can be written

$$P_* = Z_*^-(Z_*^{-\top} Z_*^-)^{-1} Z_*^{-\top}.$$

Since Z_*^- is injective and $Z_*^{-\top} Z_*^-$ is nonsingular, condition (15.16) can be written

$$Z_*^{-\top}(L_* - M_k) d_k = o(\|d_k\|) \quad \text{or} \quad (Z_*^{-\top} L_* - Z_k^{-\top} M_k) d_k = o(\|d_k\|).$$

To write the last condition, we have supposed that $Z^-(\cdot)$ is continuous at x_* and that $\{M_k\}$ is bounded. This shows that the important part of M_k is $Z_k^{-\top} M_k$, which reminds us that only the part $Z_k^{-\top} L_k$ of L_k plays a role in the definition of the Newton direction for equality constrained problems (see observation 1 on page 235).

15.4 The Hanging Chain Project III

In this third session, we resume the project on the determination of the static equilibrium position of a hanging chain, started in § 13.8 and developed in § 14.7. Our present objective is to implement the local SQP algorithm, presented on page 257, to be able to take into account the floor constraint. The algorithm is quite similar to the Newton method implemented in the second session. The main difference is that the solver of linear equations has to be replaced by a solver of quadratic optimization problems. This simple change will have several consequences that are discussed in this section.

It is a good idea to keep the work done in the second session and to use `mi` $= m_I$ as a flag that makes the `sqp` function select the type of solver (linear or quadratic), depending on the presence of inequality constraints. Solving a linear system is indeed much simpler than solving a quadratic optimization problem, so that the `sqp` function must be allowed to take advantage of the absence of inequality constraints.

Modifications to Bring to the `sqp` Function

Most of the work has been done in the previous session. There are only two modifications to bring to the function `sqp`.

The main change consists in substituting a quadratic optimization solver (to solve (15.4)) for the linear solver previously used in `sqp` (see chapter 14). Writing a solver of quadratic optimization problems is a difficult task. Fortunately, in our case, the MATLAB solver `quadprog` can be used, so that we can concentrate on other aspects of the SQP algorithm. `Quadprog` first finds an initial feasible point by solving a linear optimization problem and then uses an active set method to find a solution to the quadratic problem. It can detect infeasibility and unboundedness.

A second change deals with the determination of the initial dual solution $\lambda = (\lambda_E, \lambda_I)$. Since it is known that λ_I must be nonnegative, it is better now to determine λ as a solution to the bound constrained least-squares problem

$$\min_{\substack{\lambda=(\lambda_E,\lambda_I)\in\mathbb{R}^m \\ \lambda_I\geq 0}} \frac{1}{2}\|\nabla_x \ell(x,\lambda)\|_2^2,$$

instead of using (14.47). This convex quadratic optimization problem always has a solution (theorem 19.1). It can be solved by `quadprog`.

Checking the Correctness of the SQP Solver

There is little change to make an error on the part of the simulator dealing with the inequality constraints, since these are very simple. Nevertheless, it is better to check it and to verify the implementation of the quadratic solver.

The same strategy as in the case with equality constrained problems can be followed: trying to solve more and more difficult problems and check the quadratic convergence of the generated sequence.

Let us check the quadratic convergence on the following variant of test case 1a, in which we add a floor constraint.

Test case 1e: same data as for the test case 1a (namely second hook at $(a, b) = (1, -0.3)$ and bars of lengths $L = (0.4, 0.3, 0.25, 0.2, 0.4)$) with an additional floor with parameters $(g_0, g_1) = (-0.35, -0.2)$ (see the definition of the floor in (13.25)). The initial positions of the chain joints are $(0.1, -0.3)$, $(0.4, -0.5)$, $(0.6, -0.4)$, and $(0.7, -0.5)$.

The results obtained with test case 1e are shown in figure 15.2. Convergence

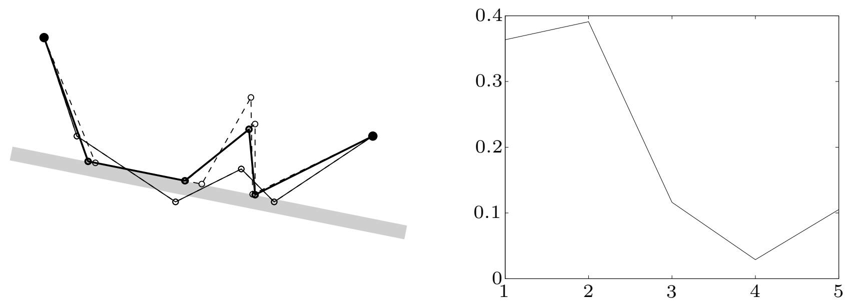

Fig. 15.2. Test case 1e

with $\texttt{options.tol(1:4)} = 10^{-10}$ is again obtained in 6 iterations. The picture on the left uses the same conventions as before: the thin solid bars represent the initial position of the chain, the dashed bars correspond to the 5 intermediate positions (hardly distinguishable), and the bold solid bars are those of the final optimal position. This one is a local minimum (the multipliers associated with the inequality constraints are positive and the critical cone is reduced to $\{0\}$). The picture on the right gives a plot of the ratios $\|s_{k+1}\|_2/\|s_k\|_2^2$, where $s_k = z_{k+1} - z_k$, for $k = 1, \ldots, 5$. The boundedness of these ratios shows without any doubt that the sequence $\{z_k\} = \{(x_k, \lambda_k)\}$ converges quadratically to its limit, as predicted by the theory (theorems 15.2 and 15.4).

Experimenting with the SQP Algorithm

A first observation, with unpleasant consequences, is that `quadprog` is aimed at computing a local minimum of a quadratic problem, not an arbitrary stationary point (note that finding a solution or a stationary point of a nonconvex quadratic problem is NP-hard, see [354] for example). Therefore, it is quite frequent to find situations where `quadprog` fails to find a stationary point, as required by the SQP algorithm. For example, with test case 1f below, which is test case 1e with the initial position of the chain given in test case 1a, the first osculating quadratic problem is unbounded.

Test case 1f: same data as for the test case 1e; but the initial positions of the chain joints are $(0.2, -0.5)$, $(0.4, -0.6)$, $(0.6, -0.8)$, and $(0.8, -0.6)$.

The unboundedness of an osculating quadratic problem can occur only when its feasible set is unbounded and the Hessian of the Lagrangian L is not positive definite at the current iterate. Hence, taking a positive definite approximation of L cures the difficulty. This can be obtained by using a quasi-Newton approximation of L; this technique is considered in chapter 18. Another possibility is add to L a (not too large) positive diagonal matrix E, such that $L+E$ is positive definite (for example by using a modified Cholesky factorization of L [154, 201]). Figure 15.3 shows the results obtained with this technique.

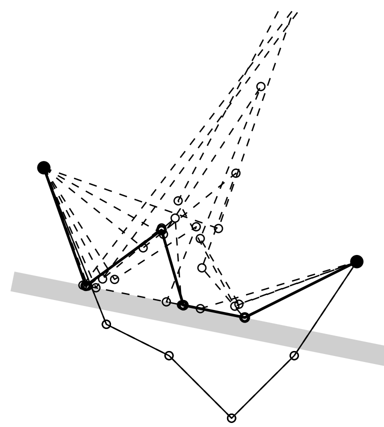

Fig. 15.3. Test case 1f

The optimal chain is actually a local minimum (the critical cone is reduced to $\{0\}$ and the energy is $e = -0.489$), different from the one obtained in figure 15.2 (in which $e = -0.518$). Observe that, although the initial position of the chain is not feasible for the floor constraint, the subsequent positions are all feasible. This is due to the affinity of the floor constraint (see (13.25) and exercise 15.1).

Another difficulty arises when the linearized constraints are incompatible, leading to an infeasible osculating quadratic problem. This difficulty is encountered at the first iteration with the initial chain given in test case 1g below. Remedies for this kind of situations exist, see [351, 341, 156] and the references thereof.

Test case 1g: same data as for the test case 1e; but the initial positions of the chain joints are $(0.1, -0.3)$, $(0.3, -0.4)$, $(0.6, -0.4)$, and $(0.7, -0.4)$.

Notes

The SQP algorithm, in a form equivalent to the one introduced in § 15.1 on page 257, was first proposed by Wilson [359; 1963]. This author was mainly concerned with the extension of the simplex method, first to quadratic programming, and then to nonlinear convex optimization problems. The algorithm was obtained by searching for a saddle point of a quadratic approximation of the Lagrangian in the primal and dual variables. No convergence proof was given. See also the introduction of this part of the book, on page 191, for other references on the origin of the SQP algorithm.

The local quadratic convergence of theorem 15.2 is due to several authors; see for example [307], in which various classes of algorithms are considered. Theorem 15.4 is taken from [38]; further refinements can be found in [40].

The criterion (15.16) for superlinear convergence dates back to Dennis and Moré [104], who introduced a similar condition to characterize the superlinear convergence of sequences generated by quasi-Newton methods in unconstrained optimization (see theorem 4.11). It was extended to problems with equality constraints by Boggs, Tolle, and Wang [36], under a somewhat strong assumption (linear convergence of the sequence $\{x_k\}$). The possibility of getting rid of this assumption has been observed by many authors. The generalization to inequality constrained problems given in theorem 15.7 is due to Bonnans [40], who uses a projector varying along the iterations; in contrast, we use the projector P_* onto the critical cone.

The local convergence of the SQP algorithm has been extended to different contexts, such as semi-infinite programming [180], infinite dimension programming [3, 4, 5, 6, 219]. When (MF-CQ) holds, but not (LI-CQ), the optimal multiplier may not be unique, so that the limit behavior of the multiplier sequence $\{\lambda_k^{\mathrm{QP}}\}$ is difficult to predict; this situation is analyzed in [367, 183, 7, 8].

Exercise

15.1. Consider the SQP algorithm applied to problem (P_{EI}) in which the ith constraint, for some $i \in E \cup I$ (equality or inequality constraint), is affine (i.e., $c_i(x+d) = c_i(x) + c_i'(x)\cdot d$ for all x and $d \in \mathbb{R}^n$). Let (x, λ) be the current iterate and define x_+ by $x_+ := x + \alpha d$, where d is a solution to the osculating quadratic problem (15.4) (we drop the index k) and $\alpha \in]0, 1]$. Show that x_+ is feasible for the ith constraint (i.e., $c_i(x_+) = 0$ if $i \in E$, or $c_i(x_+) \leq 0$ if $i \in I$) if either x is feasible for the ith constraint or if $\alpha = 1$.

16 Exact Penalization

16.1 Overview

The algorithms studied in chapters 14 and 15 generate converging sequences if the first iterate is close enough to a regular stationary point (see theorems 14.4, 14.5, 14.7, 15.2, and 15.4). Such an iterate is not necessarily at hand, so it is important to have techniques that allow the algorithms to force convergence, even when the starting point is far from a solution. This is known as the *globalization* of a local algorithm. The term is a little ambiguous, since it may suggest that it has a link with the search of global minimizers of (P_{EI}). This is not at all the case (for an entry point on *global optimization*, see [200]).

There are (at least) two classes of techniques to globalize a local algorithm: *line-search* and *trust-region*; we shall only consider the line-search approach in this survey. Both techniques use the same idea: the progress made from one iterate x_k to the next one x_{k+1} towards the solution is measured by means of an auxiliary function, called the *merit function* (the novel notion of *filter*, not discussed in this part, looks like a promising alternative; see [130] for the original paper). In unconstrained optimization, "the" appropriate merit function is of course the objective f itself. Here, the measure has to take into account the two, usually contradictory, goals in (P_{EI}): minimizing f and satisfying the constraints. Accordingly, the merit function has often the following form

$$f(x) + p(x),$$

where p is a function penalizing the constraint violation: p is zero on the feasible set and positive outside. Instead of merit functions, one also speaks of *penalty functions*, although the latter term is usually employed when the penalty function is minimized by algorithms for unconstrained optimization. As we shall see, the approach presented here is more subtle: truly constrained optimization algorithms are used (like those in chapters 14 and 15); the merit function only intervenes as a tool for measuring the adequacy of the step computed by the local methods. It is not used for computing the direction itself. The main advantage is that the ill-conditioning encountered with penalty methods is avoided, and the fast speed of convergence of the local methods is ensured close to a solution.

As many merit functions exist, a selection must be made. We shall only study those that do not use the derivatives of f and c. These are the most widely encountered in optimization codes and their numerical effectiveness has been demonstrated. To start with, let us examine some common examples of merit/penalty functions. We denote by $\|\cdot\|_2$ the ℓ_2 norm and by $\|\cdot\|_{P}$ an arbitrary norm.

(a) Quadratic penalization:

$$f(x) + \frac{\sigma}{2}\|c(x)^{\#}\|_2^2. \tag{16.1}$$

(b) Lagrangian:

$$f(x) + \mu^\top c(x).$$

(c) Augmented Lagrangian (case $I = \emptyset$):

$$f(x) + \mu^\top c(x) + \frac{\sigma}{2}\|c(x)\|_2^2. \tag{16.2}$$

Augmented Lagrangian (general case):

$$\begin{aligned} f(x) &+ \mu_E^\top c_E(x) + \frac{\sigma}{2}\|c_E(x)\|_2^2 \\ &+ \sum_{i\in I}\left(\mu_i \max\left(\frac{-\mu_i}{\sigma}, c_i(x)\right) + \frac{\sigma}{2}\left[\max\left(\frac{-\mu_i}{\sigma}, c_i(x)\right)\right]^2\right). \end{aligned} \tag{16.3}$$

(d) Nondifferentiable augmented function:

$$f(x) + \sigma\|c(x)^{\#}\|_{P}.$$

These functions have quite different features. One important property that distinguishes them is the *exactness* of the penalization, which is the subject of the present chapter. The concept of exact penalization is sometimes ambiguous – or at least varies from author to author. We adopt the following definition.

A function $\Theta : \Omega \to \mathbb{R}$ is called an *exact penalty function* at a local minimum x_* of (P_{EI}) if x_* is a local minimum of Θ. The converse implication (x_* is a local minimum of (P_{EI}) whenever it minimizes Θ locally) is not generally possible unless feasibility of x_* is assumed. The example in figure 16.1 is an illustration: x'_* is a local minimum of the functions (a) or (d) with $\sigma \geq 0$ but, being infeasible, it is not a solution to the minimization problem. The reason why the concept of exactness is so important for globalizing the SQP algorithm will be discussed in chapter 17.

Table 16.1 gives some properties of the merit functions (a)–(d). This deserves some comments.

- As far as the differentiability of Θ_σ is concerned, we assume that f and c are of class C^∞. We see that, in general, the presence of inequality constraints decreases the degree of differentiability of the merit functions. In this respect, the Lagrangian (b) is an exception.

$$\begin{cases} \min_x 0 \\ c(x) = 0 \end{cases}$$

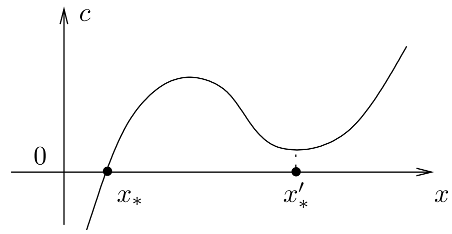

Fig. 16.1. Exactness and feasibility

Function	Differentiability	Exactness	Conditions for exactness	Threshold of σ depends on
(a)	C^1	no		
(b)	C^∞	yes	(P_{EI}) convex $\mu = \lambda_*$	
(c)	C^1	yes	$\mu = \lambda_*$ σ large	2nd derivatives
(d)	C^0	yes	σ large	1st derivatives

Table 16.1. Comparison of some merit functions

- We also see that only functions (b)–(d) can be exact. The quadratic penalty function is hardly ever exact: if $I = \emptyset$, it is differentiable and its gradient at a solution x_* is $\nabla f(x_*)$, which is usually nonzero. As we shall see in the following sections, the Lagrangian (b) is exact for convex problems and the augmented Lagrangian (c) is exact for nonconvex problems provided the penalty parameter σ is large enough.
- To be exact, both functions (b) and (c) need to have $\mu = \lambda_*$. From an algorithmic point of view, this means that the value of μ must be continually modified in order to approximate the unknown optimal multiplier λ_*. Algorithms using the Lagrangians do not minimize the same function at each iteration, which can raise convergence difficulties.
- Another shortcoming of (c) is that the threshold of σ, beyond which the penalization becomes exact, involves the eigenvalues of the Hessian of the Lagrangian. It is therefore not easily accessible to computation, and certainly out of reach if the Hessian of the Lagrangian is not explicitly computed, as in the quasi-Newton versions of the algorithms. Nevertheless, many algorithms use this function (for example, those described in [85]).

- Finally, the conditions for the exactness of function (d) are less restrictive and this is the main reason why this merit function is often used for globalizing the SQP algorithm, as in chapter 17. We shall see in particular that the threshold of σ can easily be estimated during the iterations, with the help of an estimate of the optimal multiplier. Function (d) is nonsmooth, however.

In the rest of this chapter, we study some properties of the merit functions (b)–(d), focusing on their exactness.

16.2 The Lagrangian

In this section, problem (P_{EI}) is assumed to be convex: f and the c_i's, $i \in I$, are convex, and c_E is affine. In this case, the Lagrangian of the problem is exact at a solution x_*, providing the multiplier is set to a dual solution λ_*. Actually, proposition 16.1 below shows a little more than that: ℓ has a saddle-point at (x_*, λ_*), a concept made precise in the next definition.

Let X and Y be two sets and let $\varphi : X \times Y \to \mathbb{R}$ be a function. We say that $(x_*, y_*) \in X \times Y$ is a *saddle-point* of φ on $X \times Y$ when

$$\varphi(x_*, y) \leq \varphi(x_*, y_*) \leq \varphi(x, y_*), \quad \text{for all } x \in X \text{ and } y \in Y.$$

Thus, $x \mapsto \varphi(x, y_*)$ is minimal at x_* and $y \mapsto \varphi(x_*, y)$ is maximal at y_*.

Recall that the *Lagrangian* of problem (P_{EI}) is the function

$$(x, \mu) \in \Omega \times \mathbb{R}^m \mapsto \ell(x, \mu) = f(x) + \mu^\top c(x). \tag{16.4}$$

If a feasible point x_* minimizes $\ell(\cdot, \mu)$, then $0 = \nabla_x \ell(x_*, \mu)$, which indicates that x_* will be a solution to (P_{EI}) provided μ is a dual solution. The following result shows that, for convex problems, the primal-dual solutions to (P_{EI}) are saddle-points of ℓ on $\Omega \times \{\mu \in \mathbb{R}^m : \mu_I \geq 0\}$. The way is then open to computing primal-dual solutions to (P_{EI}) with algorithms computing saddle-points. We shall not proceed in that way but it is useful to bear this point of view in mind. In addition, this result shows that $x \mapsto \ell(x, \lambda_*)$ is an exact penalty function for convex problems.

Proposition 16.1 (saddle-point of the Lagrangian). *Suppose that problem (P_{EI}) is convex, that x_* is a solution, and that f and c are differentiable at x_*. Suppose also that there exists a multiplier λ_* such that the optimality conditions (KKT) are satisfied. Then (x_*, λ_*) is a saddle-point of the Lagrangian defined in* (16.4) *on $\Omega \times \{\mu \in \mathbb{R}^m : \mu_I \geq 0\}$.*

Proof. Take $\mu \in \{\mu \in \mathbb{R}^m : \mu_I \geq 0\}$. We have

$$\begin{aligned}
\ell(x_*, \mu) &= f(x_*) + \mu_I^\top c_I(x_*) \qquad [\text{because } c_E(x_*) = 0] \\
&\leq f(x_*) \qquad [\text{because } \mu_i c_i(x_*) \leq 0 \text{ for } i \in I] \\
&= f(x_*) + \lambda_*^\top c(x_*) \qquad [\text{because } c_E(x_*) = 0 \text{ and } (\lambda_*)_I^\top c_I(x_*) = 0] \\
&= \ell(x_*, \lambda_*).
\end{aligned}$$

On the other hand, since $(\lambda_*)_I \geq 0$ and (P_{EI}) is convex, the function $x \in \Omega \mapsto \ell(x, \lambda_*)$ is convex. According to the assumptions, this function is differentiable at x_* and, in view of the optimality conditions (KKT), we have $\nabla_x \ell(x_*, \lambda_*) = 0$. We deduce that this function is minimal at x_*: $\ell(x_*, \lambda_*) \leq \ell(x, \lambda_*)$, for all $x \in \Omega$. □

16.3 The Augmented Lagrangian

The Lagrangian (16.4) is not an exact penalty function if the problem is nonconvex. For example, the nonconvex problem

$$\begin{cases} \min_x \ \log(x) \\ x \geq 1 \end{cases}$$

has the unique primal-dual solution $(x_*, \lambda_*) = (1, 1)$ and its Lagrangian $\ell(x, \lambda_*) = \log(x) + 1 - x$ is concave with a *maximum* at $x = 1$.

The augmented Lagrangian ℓ_r obviates this shortcoming. In fact we shall prove a local version of proposition 16.1: if $\mu = \lambda_*$ and r is large enough, $\ell_r(\cdot, \mu)$ has a strict local minimum at a strong solution to the optimization problem (P_{EI}).

The augmented Lagrangian is best introduced by using a perturbation technique as in duality theory, but this is beyond the scope of this review. Here we follow a more intuitive approach, starting with the case where only equality constraints are present. In this case, one takes

$$\ell_r(x, \mu) = f(x) + \mu_E^\top c_E(x) + \frac{r}{2}\|c_E(x)\|_2^2. \tag{16.5}$$

This is the standard Lagrangian ℓ, augmented by the term $(r/2)\|c_E(x)\|_2^2$. This term penalizes the constraint violation and makes $\ell_r(\cdot, \mu)$ convex around the point x_* in a subspace complementary to the tangent space $N(A_E(x_*))$. This creates a basin around a strong solution to (P_E), making the penalization exact (this point of view is developed in exercise 16.2).

To deal with inequality constraints, we first transform (P_{EI}) by introducing slack variables $s \in \mathbb{R}^{m_I}$:

$$\begin{cases} \min_{(x,s)} \ f(x) \\ c_E(x) = 0 \\ c_I(x) + s = 0 \\ s \geq 0. \end{cases}$$

Next, this problem is *approached* by using the augmented Lagrangian associated with its equality constraints:

$$\min_x \min_{s \geq 0} \left(f(x) + \mu_E^\top c_E(x) + \frac{r}{2}\|c_E(x)\|_2^2 + \mu_I^\top (c_I(x){+}s) + \frac{r}{2}\|c_I(x){+}s\|_2^2 \right).$$

The augmented Lagrangian associated with (P_{EI}) is the function of x and μ defined by the minimal value of the optimization problem in $s \geq 0$ above:

$$\ell_r(x,\mu) := \min_{s\geq 0} \Big(f(x) + \mu_E^\top c_E(x) + \frac{r}{2}\|c_E(x)\|_2^2 \\ + \mu_I^\top (c_I(x)+s) + \frac{r}{2}\|c_I(x)+s\|_2^2 \Big).$$

Actually, the minimization in s can be carried out explicitly since the minimized function of s is quadratic with a positive diagonal Hessian. More precisely, discarding terms independent of s, the objective can be written $\frac{r}{2}\|s+c_I(x)+\mu_I/r\|_2^2$, so that the minimizer is the projection of $-c_I(x)-\mu_I/r$ on the positive orthant, namely $s = \max(-c_I(x) - \mu_I/r, 0)$. Adding $c_I(x)$, one finds

$$c_I(x) + s = \max\left(\frac{-\mu_I}{r}, c_I(x)\right).$$

Substituting $c_I(x) + s$ by this value in the objective of the problem above yields an explicit formula for the *augmented Lagrangian*. This is the function $\ell_r : \Omega \times \mathbb{R}^m \to \mathbb{R}$, defined for $(x,\mu) \in \Omega \times \mathbb{R}^m$ and $r \in \mathbb{R}_+^* := \{t \in \mathbb{R} : t > 0\}$ by

$$\ell_r(x,\mu) = f(x) + \mu^\top \tilde{c}_r(x,\mu) + \frac{r}{2}\|\tilde{c}_r(x,\mu)\|_2^2, \tag{16.6}$$

where $\tilde{c}_r : \Omega \times \mathbb{R}^m \to \mathbb{R}^m$ is defined by

$$(\tilde{c}_r(x,\mu))_i = \begin{cases} c_i(x) & \text{if } i \in E \\ \max\left(\frac{-\mu_i}{r}, c_i(x)\right) & \text{if } i \in I. \end{cases} \tag{16.7}$$

The coefficient r is called the *augmentation parameter*. This augmented Lagrangian (16.6) has therefore a structure very similar to the one associated with the equality constraint problem (P_E), see (16.5), with c_E substituted by the non-differentiable function $\tilde{c}_r$ introduced above.

Despite the nonsmoothness of the max operator in (16.7), the augmented Lagrangian is differentiable in x, provided that f and c have that property. The easiest way of verifying this claim is to write the terms associated with the inequalities in (16.6) as follows

$$\mu_I^\top (\tilde{c}_r(x,\mu))_I + \frac{r}{2}\|(\tilde{c}_r(x,\mu))_I\|_2^2 = \frac{1}{2r}\sum_{i\in I}\left(\max(0,\mu_i + rc_i(x))^2 - \mu_i^2\right).$$

This is a differentiable function of x, since $\max(0,\cdot)$ is squared. A straightforward computation then leads to

$$\nabla_x \ell_r(x,\mu) = \nabla f(x) + c'(x)^\top(\mu + r\tilde{c}_r(x,\mu)). \tag{16.8}$$

Second differentiability in x is also ensured around a primal solution satisfying some strong conditions. Let x_* be a solution to (P_{EI}) and let λ_*

be a multiplier associated with x_*. Using the complementarity conditions $(\lambda_*)_I^\top c_I(x_*) = 0$ and the nonnegativity of $(\lambda_*)_I$, it is not difficult to see that, *for x close to x_**, there holds

$$\ell_r(x,\lambda_*) = \ell(x,\lambda_*) + \frac{r}{2}\sum_{i\in E\cup I_*^{0+}} c_i(x)^2 + \frac{r}{2}\sum_{i\in I_*^{00}} (c_i(x)^+)^2. \tag{16.9}$$

Because of the operator $(\cdot)^+$ in (16.9), $\ell_r(\cdot,\lambda_*)$ may not be twice differentiable at x_*. In the case of *strict complementarity*, however, $I_*^{00} = \emptyset$ and the last sum disappears, so that the augmented Lagrangian can be written (for x close to x_*)

$$\ell_r(x,\lambda_*) = \ell(x,\lambda_*) + \frac{r}{2}\sum_{i\in E\cup I_*^{0}} c_i(x)^2.$$

Locally, equality and active inequality constraints are then treated in the same way and $\ell_r(\cdot,\lambda_*)$ is smooth around x_* (provided f and c are smooth). The next proposition gathers these differentiability properties.

Proposition 16.2 (differentiability of the augmented Lagrangian). *If f and c are differentiable at x, then the augmented Lagrangian ℓ_r, defined by* (16.6)*, is differentiable at x and its gradient is given by* (16.8)*. If (x_*,λ_*) is a KKT point for (P_{EI}) satisfying strict complementarity and if $(f, c_{E\cup I_*^0})$ is p times differentiable* (*with $p \geq 0$ integer*) *in some neighborhood of x_*, then the augmented Lagrangian is p times differentiable is some* (*possibly smaller*) *neighborhood of x_*.*

The next result gives conditions for (x_*,λ_*) to be a saddle-point of ℓ_r on $V\times\mathbb{R}^m$, where V is a neighborhood of x_* in Ω. Compared with proposition 16.1, the result is local in x, but global in μ, and the minimum in x is strict. As before, this result implies that, if r is large enough (but finite!), $\ell_r(\cdot,\lambda_*)$ is an exact penalty function for (P_{EI}).

Proposition 16.3 (saddle-point of the augmented Lagrangian). *Suppose that f and $c_{E\cup I_*^0}$ are twice differentiable at a local minimum x_* of (P_{EI}) at which the KKT conditions hold, and that the semi-strong second-order sufficient condition of optimality* (13.9) *is satisfied for some multiplier λ_*. Then there exist a neighborhood V of x_* in Ω and a number $\underline{r} > 0$ such that, for all $r \geq \underline{r}$, (x_*,λ_*) is a saddle-point of ℓ_r on $V\times\mathbb{R}^m$. More precisely, we have for all $(x,\mu)\in(V\backslash\{x_*\})\times\mathbb{R}^m$:*

$$\ell_r(x_*,\mu) \leq \ell_r(x_*,\lambda_*) < \ell_r(x,\lambda_*). \tag{16.10}$$

Proof. Let us first show that λ_* maximizes $\ell_r(x_*,\cdot)$ for any $r > 0$. We have for $\mu\in\mathbb{R}^m$:

$$\ell_r(x_*,\mu) = f(x_*) + \sum_{\substack{i\in I\\ c_i(x_*)\geq -\mu_i/r}} \Big(\mu_i c_i(x_*) + \frac{r}{2}c_i(x_*)^2\Big) - \sum_{\substack{i\in I\\ c_i(x_*)< -\mu_i/r}} \frac{\mu_i^2}{2r}.$$

The maximum in μ can be obtained term by term. If $c_i(x_*) = 0$, the maximum in the right-hand side is $f(x_*)$, obtained for all $\mu_i \geq 0$. If $c_i(x_*) < 0$, this maximum is again $f(x_*)$, obtained for $\mu_i = 0$. Since $(\lambda_*)_I$ satisfies these conditions, we have

$$\ell_r(x_*, \mu) \leq f(x_*) = \ell_r(x_*, \lambda_*), \quad \text{for all } \mu \in \mathbb{R}^m.$$

Let us now show the second statement, dealing with the *strict* local minimality of x_*. Note that we need to prove the inequality on the right in (16.10) for only a single value of r, $\underline{r} > 0$ say, because then, this inequality will hold for any $r \geq \underline{r}$ and any $x \in V$ (independent of r). Indeed, $\ell_r(x_*, \lambda_*) = f(x_*)$ does not depend on r and, for fixed x, $r \mapsto \ell_r(x, \lambda_*)$ is nondecreasing (this is a clear consequence of the way the augmented Lagrangian was introduced, just before the proposition).

We prove this by contradiction, assuming that there is a sequence of positive numbers $r_k \to \infty$ and a sequence of points $x_k \to x_*$, with $x_k \neq x_*$ such that, for $k \geq 1$:

$$\ell_{r_k}(x_k, \lambda_*) \leq \ell_{r_k}(x_*, \lambda_*). \tag{16.11}$$

Taking a subsequence if necessary, we have for $k \to \infty$:

$$\frac{x_k - x_*}{\|x_k - x_*\|} \to d, \quad \text{with } \|d\| = 1.$$

Hence, setting $\alpha_k := \|x_k - x_*\|$, we have

$$x_k = x_* + \alpha_k d + o(\alpha_k).$$

Our aim now is to show that d is a critical direction. We do this by appropriate expansions in the left-hand side of (16.11): second order expansion of the Lagrangian and first order expansion of the constraints in both sums of (16.9). To simplify the notation, we introduce $L_* = \nabla^2_{xx}\ell(x_*, \lambda_*)$. From the smoothness of f and c and the optimality of (x_*, λ_*), we have

$$\begin{aligned} \ell(x_k, \lambda_*) &= \ell(x_*\lambda_*) + \frac{\alpha_k^2}{2}\, d^\top L_* d + o(\alpha_k^2), \\ c_i(x_k) &= \alpha_k\, c_i'(x_*) \cdot d + o(\alpha_k), \quad \text{for } i \in E \cup I_*^0. \end{aligned}$$

Injecting these estimates in (16.11), using (16.9) and $\ell_{r_k}(x_*, \lambda_*) = \ell(x_*, \lambda_*)$, provides

$$\begin{aligned} \frac{\alpha_k^2}{2}\, d^\top L_* d + o(\alpha_k^2) + \frac{r_k}{2} \sum_{i \in E \cup I_*^{0+}} &\left(\alpha_k c_i'(x_*) \cdot d + o(\alpha_k)\right)^2 \\ &+ \frac{r_k}{2} \sum_{i \in I_*^{00}} \left([\alpha_k c_i'(x_*) \cdot d + o(\alpha_k)]^+\right)^2 \leq 0. \end{aligned} \tag{16.12}$$

Dividing by $\alpha_k^2 r_k$ and taking the limit yield

$$\begin{aligned} c_i'(x_*)\cdot d &= 0, \quad \text{if } i \in E \cup I_*^{0+} \\ c_i'(x_*)\cdot d &\le 0, \quad \text{if } i \in I_*^{00}. \end{aligned}$$

Therefore d is a nonzero critical direction.

On the other hand, (16.12) also implies that

$$\frac{\alpha_k^2}{2}\, d^\top L_* d + o(\alpha_k^2) \le 0.$$

Dividing by α_k^2 and taking the limit show that $d^\top L_* d \le 0$, which contradicts assumption (13.9) since $d \in C_*\backslash\{0\}$. □

In the previous result, the semi-strong second-order sufficient condition of optimality (13.9) is assumed. If only the weak condition (13.8) holds, $\ell_r(\cdot,\lambda_*)$ may not have a local minimum at x_*, whatever the choice of $\lambda_* \in \Lambda_*$ and the value of r. An example is given in exercise 16.4.

16.4 Nondifferentiable Augmented Function

We now consider the following merit function for problem (P_{EI}):

$$\Theta_\sigma(x) = f(x) + \sigma \|c(x)^\#\|_P, \tag{16.13}$$

which we call the *nondifferentiable augmented function*. In (16.13), $\sigma > 0$ is called the *penalty parameter*, the operator $(\cdot)^\#$ was defined on page 194, and $\|\cdot\|_P$ is a norm, and is arbitrary for the moment. We denote by $\|\cdot\|_D$ the *dual norm* of $\|\cdot\|_P$, with respect to the Euclidean scalar product. It is defined by

$$\|v\|_D = \sup_{\|u\|_P = 1} v^\top u.$$

We therefore have the *generalized Cauchy-Schwarz inequality*:

$$|u^\top v| \le \|u\|_P \|v\|_D, \quad \text{for all } u \text{ and } v. \tag{16.14}$$

See exercise 16.5 for some examples of dual norms.

Because of the norm $\|\cdot\|_P$ and of the operator $(\cdot)^\#$, Θ_σ is usually nondifferentiable; but when f and c are smooth, Θ_σ has directional derivatives; this is a consequence of lemma 13.1. It so happens that this differentiability concept will be sufficient for our development.

Let $v \in \mathbb{R}^m$ be such that $v_I \le 0$ and denote by $P_v : \mathbb{R}^m \to \mathbb{R}^m$ the operator defined by $P_v u = (\cdot^\#)'(v;u)$, that is

$$(P_v u)_i = \begin{cases} u_i & \text{si } i \in E \\ u_i^+ & \text{if } i \in I \text{ and } v_i = 0 \\ 0 & \text{if } i \in I \text{ and } v_i < 0. \end{cases}$$

This notation allows us to write concisely the directional derivative of Θ_σ at a feasible point.

Lemma 16.4. *If f and c have a directional derivative at x in the direction $h \in \mathbb{R}^n$, then Θ_σ has also a directional derivative at x in the direction h. If, in addition, x is feasible for (P_{EI}), we have*

$$\Theta'_\sigma(x;h) = f'(x;h) + \sigma \|P_{c(x)} c'(x;h)\|_{_P}.$$

Proof. The directional differentiability of $\Theta_\sigma = f + \sigma(\|\cdot\|_{_P} \circ (\cdot)^\# \circ c)$ comes from lemma 13.1, the assumptions on f and c, and the fact that $(\cdot)^\#$ and $\|\cdot\|_{_P}$ are Lipschitzian and have directional derivatives.

If x is feasible, $c(x)^\# = 0$ and we have from lemma 13.1,

$$\Theta'_\sigma(x;h) = f'(x;h) + \sigma(\|\cdot\|_{_P})'(0;(c^\#)'(x;h)).$$

On the other hand,

$$(c^\#)'(x;h) = (\cdot^\#)'(c(x);c'(x;h)) = P_{c(x)} c'(x;h)$$

and

$$(\|\cdot\|_{_P})'(0;v) = \lim_{t\to 0+} \frac{1}{t}(\|tv\|_{_P} - 0) = \|v\|_{_P}.$$

The result follows. $\square$

Necessary Conditions of Exactness

In this subsection, we examine which conditions are implied by the fact that a *feasible* point x_* is a minimum point of Θ_σ. We quote three such properties in proposition 16.5: x_* is also a minimum point of (P_{EI}), there exists a multiplier λ_* associated with x_*, and σ must be sufficiently large. The second property shows that the exactness of Θ_σ plays a similar role as a constraint qualification assumption, since it implies the existence of a dual solution.

For the third property mentioned above, we need an assumption on the norm $\|\cdot\|_{_P}$ used in Θ_σ. The norm $\|v\|_{_P}$ must decrease if one sets to zero some of the I-components of $v \in \mathbb{R}^m$:

$$u_i = \begin{cases} v_i & \text{if } i \in E \\ 0 \text{ or } v_i & \text{if } i \in I \end{cases} \quad \Longrightarrow \quad \|u\|_{_P} \le \|v\|_{_P}. \tag{16.15}$$

Clearly, ℓ_p norms, $1 \le p \le \infty$, satisfy this property; but it is not necessarily satisfied by an arbitrary norm (see exercise 17.1). Also, the claim on σ in proposition 16.5 may not be correct if $\|\cdot\|_{_P}$ does not satisfy (16.15) (see exercise 16.6).

Proposition 16.5 (necessary conditions of exactness). *If x_* is feasible for (P_{EI}) and Θ_σ has a local minimum (resp. strict local minimum) at x_*, then x_* is a local minimum (resp. strict local minimum) of (P_{EI}). If, in addition, f and c are Gâteaux differentiable at x_*, then there exists a multiplier λ_* such that the necessary optimality conditions (KKT) hold. If, in addition, the norm $\|\cdot\|_{_P}$ satisfies (16.15) and (LI-CQ) holds at x_*, then $\sigma \ge \|\lambda_*\|_{_D}$.*

Proof. If x_* is a local minimum of Θ_σ, there exists a neighborhood V of x_* such that

$$\Theta_\sigma(x_*) \le \Theta_\sigma(x), \quad \text{for all } x \in V.$$

Since $x_* \in X$ and $\Theta_\sigma|_X = f|_X$, we have

$$f(x_*) \le f(x), \quad \text{for all } x \in V \cap X,$$

which shows that x_* is a local minimum of (P_{EI}). The above inequality is strict for $x \ne x_*$, if x_* is a strict local minimum of Θ_σ.

Now suppose f and g are Gâteaux differentiable at x_*. Then Θ_σ has directional derivatives at x_* (lemma 16.4). Since x_* is a local minimum of Θ_σ, we have $\Theta'_\sigma(x_*; h) \ge 0$ for all $h \in \mathbb{R}^m$. But x_* is feasible; hence, by lemma 16.4:

$$\nabla f(x_*)^\top h + \sigma \| P_{c(x_*)}(A(x_*)h)\|_{P} \ge 0, \quad \text{for all } h \in \mathbb{R}^m. \tag{16.16}$$

We deduce

$$P_{c(x_*)}(A(x_*)h) = 0 \quad \Longrightarrow \quad \nabla f(x_*)^\top h \ge 0.$$

Thus, $h = 0$ solves the linear program

$$\begin{cases} \min_h \ \nabla f(x_*)^\top h \\ A_E(x_*)h = 0, \\ A_{I^0_*}(x_*)h \le 0. \end{cases}$$

The constraints of this problem being qualified (by (A-CQ)), we deduce the existence of a multiplier $\lambda_* \in \mathbb{R}^m$ such that

$$\begin{cases} \nabla f(x_*) + A(x_*)^\top \lambda_* = 0 \\ (\lambda_*)_{I^0_*} \ge 0 \\ (\lambda_*)_{I \setminus I^0_*} = 0. \end{cases}$$

Since x_* is feasible, (KKT) holds with (x_*, λ_*).

Finally, suppose that the norm $\|\cdot\|_{P}$ satisfies (16.15) and that (LI-CQ) holds. Take again (16.16) and use the first-order optimality condition to obtain

$$\lambda_*^\top A(x_*)h \le \sigma \|P_{c(x_*)}A(x_*)h\|_{P}, \quad \text{for all } h \in \mathbb{R}^n.$$

Set $J = E \cup I^0_*$, and remember that $(\lambda_*)_i = 0$ if $i \notin J$. For an arbitrary v in $\mathbb{R}^m$, we have $\lambda_*^\top v = (\lambda_*)_J^\top v_J$ and, from (LI-CQ), we can find $h \in \mathbb{R}^n$ such that $A_J(x_*)h = v_J$. We deduce that

$$\lambda_*^\top v = (\lambda_*)_J^\top A_J(x_*)h = \lambda_*^\top A(x_*)h \le \sigma \|P_{c(x_*)}A(x_*)h\|_{P} \le \sigma \|v\|_{P},$$

where the last inequality uses property (16.15) of the norm. Then $\lambda_*^\top v \le \sigma \|v\|_{P}$, and since v is arbitrary, we have $\|\lambda_*\|_{D} \le \sigma$. $\square$

Sufficient Conditions of Exactness

In practice, we are more interested in having conditions that ensure the exactness of Θ_σ and this is what we focus on now. We shall show that the necessary condition obtained on σ in proposition 16.5 is sharp: if x_* is a strong solution to problem (P_{EI}) with associated multiplier λ_*, x_* also minimizes Θ_σ provided $\sigma > \|\lambda_*\|_D$ (the strict inequality is not needed for convex problems). This result holds without any particular assumption on the norm $\|\cdot\|_P$.

The necessary conditions of exactness of Θ_σ were obtained by expressing the fact that, if x_* minimizes Θ_σ, the directional derivative $\Theta'_\sigma(x_*;h)$ must be nonnegative for all $h \in \mathbb{R}^n$ (see the proof of proposition 16.5). Now we want to exhibit values of σ such that Θ_σ has a minimum at x_*. Function Θ_σ is nondifferentiable and nonconvex. Therefore, it is not sufficient to show that $\Theta'_\sigma(x_*;h) \geq 0$ for all $h \in \mathbb{R}^n$ in order to ensure its exactness. One cannot impose $\Theta'_\sigma(x_*;h) > 0$ for all $h \in \mathbb{R}^n$ either, since this may never occur for any value of σ (for example, when $E \neq \emptyset$ and $I = \emptyset$, $\Theta'_\sigma(x_*;h) = 0$ for any h in the space tangent to the constraint manifold). Therefore, we shall use either a technical detour (for convex problems) or a direct proof like the one of proposition 16.3 (for nonconvex problems).

In proposition 16.7 below, we consider the case of convex problems and in proposition 16.8 the case of nonconvex problems. To prove the exactness of the nondifferentiable function Θ_σ for convex problems, we simply use the fact that, if σ is large enough, Θ_σ is above the differentiable Lagrangian (16.4) (lemma 16.6), which is known to be exact at x_* (proposition 16.1). Observe that lemma 16.6 does not assume convexity.

Lemma 16.6. *If $\sigma \geq \|\lambda\|_D$ and $\lambda_I \geq 0$, then $\ell(\cdot,\lambda) \leq \Theta_\sigma(\cdot)$ on $\mathbb{R}^n$.*

Proof. First observe that $\lambda_I \geq 0$ implies $\lambda_I^\top c_I(x) \leq \lambda_I^\top c_I(x)^+$. Then, for all $x \in \mathbb{R}^n$,

$$\ell(x,\lambda) \leq f(x) + \lambda^\top c(x)^\# \leq f(x) + \|\lambda\|_D \|c(x)^\#\|_P \leq \Theta_\sigma(x).$$

$\square$

Proposition 16.7 (sufficient conditions of exactness, convex problems). *Suppose that problem (P_{EI}) is convex and that f and c are differentiable at a solution x_* to (P_{EI}) with an associated multiplier λ_*. Then Θ_σ has a global minimum at x_* as soon as $\sigma \geq \|\lambda_*\|_D$.*

Proof. According to proposition 16.1, $\ell(\cdot,\lambda_*)$ is minimized by x_* and, by lemma 16.6, it is dominated by Θ_σ ($\sigma \geq \|\lambda_*\|_D$ and $(\lambda_*)_I \geq 0$). Therefore

$$\begin{aligned}
\Theta_\sigma(x_*) &= f(x_*) \\
&= \ell(x_*,\lambda_*) \\
&\leq \ell(x,\lambda_*), \quad \text{for all } x \in \mathbb{R}^n \\
&\leq \Theta_\sigma(x), \quad \text{for all } x \in \mathbb{R}^n.
\end{aligned}$$

$\square$

The same technical detour could be used for highlighting sufficient conditions of exactness of Θ_σ for nonconvex problems: if $\sigma > \|\lambda_*\|_{_D}$, Θ_σ is above the augmented Lagrangian (16.6) in some neighborhood of x_*, so that the exactness of Θ_σ follows that of the augmented Lagrangian (proposition 16.3). This strategy is proposed in exercise 16.7. The direct proof given below has the advantage of being valid even when only the weak second order sufficient condition of optimality (13.8) holds at x_* (in contrast, the semi-strong condition (13.9) is assumed in proposition 16.3 and exercise 16.7).

Proposition 16.8 (sufficient conditions of exactness). *Suppose that f and $c_{E\cup I^0_*}$ are twice differentiable at a local minimum x_* of (P_{EI}) at which the KKT conditions hold, that the weak second-order sufficient condition of optimality* (13.8) *is satisfied, and that*

$$\sigma > \sup_{\lambda_*\in\Lambda_*} \|\lambda_*\|_{_D},$$

where Λ_* is the nonempty set of multipliers associated with x_*. Then Θ_σ has a strict local minimum at x_*.

Proof. We prove the result by contradiction, assuming that x_* is not a strict minimum of Θ_σ. Then, there exists a sequence $\{x_k\}$ such that $x_k \neq x_*$, $x_k \to x_*$ and

$$\Theta_\sigma(x_k) \leq \Theta_\sigma(x_*), \quad \forall k \geq 1. \tag{16.17}$$

Since the sequence $\{(x_k - x_*)/\|x_k - x_*\|\}$ is bounded (here $\|\cdot\|$ denotes an arbitrary norm), it has a subsequence such that $(x_k - x_*)/\|x_k - x_*\| \to d$, where $\|d\| = 1$. Denoting $\alpha_k = \|x_k - x_*\|$, one has

$$x_k = x_* + \alpha_k d + o(\alpha_k).$$

Because Θ_σ is Lipschitzian in a neighborhood of x_*:

$$\Theta_\sigma(x_k) = \Theta_\sigma(x_* + \alpha_k d) + o(\alpha_k).$$

Now (16.17) shows that $\Theta_\sigma'(x_*; d) \leq 0$. Then, from lemma 16.4, one can write

$$f'(x_*)\cdot d + \sigma\|P_{c(x_*)}(c'(x_*)\cdot d)\|_{_P} \leq 0. \tag{16.18}$$

This certainly implies that

$$f'(x_*)\cdot d \leq 0. \tag{16.19}$$

On the other hand, from the assumptions, there is an optimal multiplier λ_* such that $\sigma > \|\lambda_*\|_{_D}$. Using the first order optimality conditions, including the nonnegativity of $(\lambda_*)_I$ and the complementarity conditions $(\lambda_*)_I^\top c_I(x_*) = 0$, one has

$$\begin{aligned} -f'(x_*)\cdot d &= \lambda_*^\top (c'(x_*)\cdot d) \\ &\leq \lambda_*^\top P_{c(x_*)}(c'(x_*)\cdot d) \\ &\leq \|\lambda_*\|_{_D} \|P_{c(x_*)}(c'(x_*)\cdot d)\|_{_P}. \end{aligned}$$

Then (16.18) and $\sigma > \|\lambda_*\|_{_D}$ imply that $P_{c(x_*)}(c'(x_*)\cdot d) = 0$, i.e.,

$$\begin{cases} c_i'(x_*)\cdot d = 0 \text{ for } i \in E \\ c_i'(x_*)\cdot d \leq 0 \text{ for } i \in I_*^0. \end{cases}$$

These and (16.19) show that d is a nonzero critical direction.

Now, let λ_* be the multiplier depending on d, determined by the weak second-order sufficient condition of optimality (13.8). According to theorem 13.4, one has

$$d^\top \nabla_{xx}^2 \ell(x_*, \lambda_*) d > 0.$$

The following Taylor expansion (use $\nabla_x \ell(x_*, \lambda_*) = 0$)

$$\ell(x_k, \lambda_*) = \ell(x_*, \lambda_*) + \frac{\alpha_k^2}{2} d^\top \nabla_{xx}^2 \ell(x_*, \lambda_*) d + o(\alpha_k^2)$$

allows us to see that, for k large enough,

$$\ell(x_k, \lambda_*) > \ell(x_*, \lambda_*). \tag{16.20}$$

Then, for large indices k, there holds

$$\begin{aligned} \Theta_\sigma(x_k) &\leq \Theta_\sigma(x_*) \qquad [\text{by } (16.17)] \\ &= f(x_*) \\ &= \ell(x_*, \lambda_*) \\ &< \ell(x_k, \lambda_*) \qquad [\text{by } (16.20)] \\ &\leq \Theta_\sigma(x_k) \qquad [\text{by lemma 16.6 and } \sigma \geq \|\lambda_*\|_{_D}], \end{aligned}$$

which is the expected contradiction. □

Notes

The augmented Lagrangian (16.2) for equality constrained problems was first proposed by Arrow and Solow [14; 1958]. Hestenes [191; 1969] and Powell [288; 1969] both used this function to introduce the so-called *method of multipliers*, which has popularized this type of penalization. The augmented Lagrangian (16.3) or (16.6), adapted to inequality constrained problems, was proposed by Rockafellar [310, 311; 1971-74] and Buys [62; 1972]. It was further extended to constraints of the form $c(x) \in K$, where c is a vector-valued function and

K is a closed convex cone, by Shapiro and Sun [330; 2004]. This penalty function is usually no more than continuously differentiable, even if the problem data are infinitely differentiable. Many developments have been carried out to overcome this drawback, proposing augmentation terms with a different structure (for entry points see [17, 16; 1999-2000], which deal with primal penalty functions, and [109, 110, 111; 1999-2001], which consider primal-dual penalty functions). Surveys on the augmented Lagrangian can be found in [26, 169].

The exact penalty function (16.13) goes back at least to Eremin [119; 1966] and Zangwill [374; 1967]. Its connection with problem (P_{EI}) has been studied by many authors, see Pietrzykowski [284], Charalambous [74], Ioffe [198], Han and Mangasarian [186], Bertsekas [26], Fletcher [126], Bonnans [39, 41], Facchinei [120], Burke [60], Pshenichnyj [301], Bonnans and Shapiro [50], and the references therein.

Exercises

16.1. *Finsler's lemma* [123] *and its limit case* [9]. Let M be an $n \times n$ symmetric matrix that is positive definite on the null space of a matrix A (i.e., $u^\top M u > 0$ for all nonzero $u \in N(A)$). Show that there exists an $r_0 \in \mathbb{R}$ such that, for all $r \geq r_0$, $M + rA^\top A$ is positive definite.

[*Hint*: Use an argument by contradiction.]

Suppose now that the symmetric matrix M is only positive semidefinite on the null space of A (i.e., $u^\top M u \geq 0$ for all $u \in N(A)$). Show that the following claims are equivalent: (i) $v \in N(A)$ and $v^\top M v = 0$ imply that $Mv = 0$, and (ii) there exists an $r_0 \in \mathbb{R}$ such that, for all $r \geq r_0$, $M + rA^\top A$ is positive semidefinite. Find a matrix M that is positive semidefinite on the null space of A, for which these properties (i) and (ii) are not satisfied.

[*Hint*: For $(i) \Rightarrow (ii)$, use with care an argument by contradiction.]

Consequence: If M is nonsingular and positive semidefinite (but not positive definite) on the null space of A, it cannot enjoy property (ii) (since (i) does not hold).

16.2. *Augmented Lagrangian for equality constrained problems.* Consider problem (P_E) with functions f and c of class C^2 and the associated augmented Lagrangian $\ell_r(x, \lambda) = f(x) + \lambda^\top c(x) + \frac{r}{2}\|c(x)\|_2^2$. By a direct computation of $\nabla_x \ell_r(x_*, \lambda_*)$ and $\nabla^2_{xx} \ell_r(x_*, \lambda_*)$, show that, if r is large enough, $\ell_r(\cdot, \lambda_*)$ has a strict local minimum at a point x_* satisfying $(SC2)$.

[*Hint*: Use Finsler's lemma (exercise 16.1).]

16.3. *Fletcher's exact penalty function* [124]. Consider problem (P_E), in which f and c are smooth, and c is a submersion. Denote by $A^-(x)$ a right inverse of the constraint Jacobian $A(x) := c'(x)$ and assume that A^- is a smooth function of x. Let $\lambda^{\text{LS}}(x) := -A^-(x)^\top \nabla f(x)$ be the associated least-squares multiplier. For $r \in \mathbb{R}$, consider the function $\varphi_r : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\varphi_r(x) = f(x) + \lambda^{\text{LS}}(x)^\top c(x) + \frac{r}{2}\|c(x)\|_2^2. \tag{16.21}$$

Let (x_*, λ_*) be a pair satisfying the second-order sufficient conditions of optimality $(SC2)$ of problem (P_E). Show that there exists an $r_0 \in \mathbb{R}$, such that for $r \geq r_0$, φ_r has a strict local minimum at x_*.

[*Hint*: Prove the following claims, in which $A_* := A(x_*)$, $A_*^- := A^-(x_*)$, and $L_* := \nabla^2_{xx}\ell(x_*, \lambda_*)$, and conclude: (i) $\lambda^{\text{LS}}(x_*) = \lambda_*$; (ii) $\nabla\varphi_r(x_*) = 0$; (iii) $(\lambda^{\text{LS}})'(x_*) = -A_*^{-\top}L_*$ and $\nabla^2\varphi_r(x_*) = L_* - (A_*^\top A_*^{-\top}L_* + L_*A_*^-A_*) + rA_*^\top A_*$; (iv) $\nabla^2\varphi_r(x_*)$ is positive definite if r is large enough.]

16.4. *Counter-example for proposition* 16.3. Consider the problem in $\mathbb{R}^3$:

$$\begin{cases} \min_x \ x_3 \\ x_3 \geq (x_1 + x_2)(x_1 - x_2) \\ x_3 \geq (x_2 + 3x_1)(2x_2 - x_1) \\ x_3 \geq (2x_2 + x_1)(x_2 - 3x_1). \end{cases}$$

Show that: (i) $x_* = 0$ is the unique solution to the problem and that the associated multiplier set is $\Lambda_* = \{\lambda \in \mathbb{R}^3_+ : \lambda_1 + \lambda_2 + \lambda_3 = 1\}$; (ii) the weak second order sufficient condition of optimality (13.8) is satisfied, but not the semi-strong ones (13.9); (iii) for any $\lambda_* \in \Lambda_*$ and $r \geq 0$, the augmented Lagrangian (16.6) has not a minimum at x_*.

Consequence: When the semi-strong second order sufficient conditions of optimality (13.9) do not hold at x_*, the augmented Lagrangian $\ell_r(\cdot, \lambda_*)$ function may not have a local minimum at x_*, for any $\lambda_* \in \Lambda_*$ and $r \geq 0$.

16.5. *Dual norms.* (i) The ℓ_p norm on $\mathbb{R}^n$ is defined by

$$\|u\|_p := \begin{cases} \left(\displaystyle\sum_{i=1}^n |u_i|^p\right)^{\frac{1}{p}} & \text{if } 1 \leq p < \infty \\ \displaystyle\max_{1 \leq i \leq n} |u_i| & \text{if } p = \infty. \end{cases}$$

Show that the dual norm of $\|\cdot\|_p$ is the norm $\|\cdot\|_{p'}$, where p' is uniquely defined by

$$\frac{1}{p} + \frac{1}{p'} = 1.$$

(ii) Let Q be a symmetric positive definite matrix and define the norm $\|u\|_P = (u^\top Qu)^{\frac{1}{2}}$. Show that its dual norm is given by $\|v\|_D = (v^\top Q^{-1}v)^{\frac{1}{2}}$.

16.6. *Counter-example for proposition* 16.5. Consider the problem

$$\min\left\{\frac{1}{2}\|x\|_2^2 : x \in \mathbb{R}^2,\ x_1 \leq 0,\ x_2 + 1 \leq 0\right\}.$$

Show that the unique primal-dual solution to this problem is $x_* = (0, -1)$ and $\lambda_* = (0, 1)$. Show that $x \mapsto \|x\|_P = (x_1^2 + x_2^2 + \sqrt{3}x_1x_2)^{1/2}$ is a norm that does not satisfy (16.15), and that $\|\lambda_*\|_D = 2$. Show that $\Theta_\sigma(x) = \frac{1}{2}\|x\|_2^2 + \sigma\|(x_1, x_2 + 1)^+\|_P$ has a minimum at x_* when $\sigma \geq 1$.

Consequence: The exactness of Θ_σ does not imply $\sigma \geq \|\lambda_*\|_D$ if the norm $\|\cdot\|_P$ does not satisfy (16.15).

16.7. *A variant of proposition* 16.8. (i) Let x_* be feasible for (P_{EI}) and $\lambda \in \mathbb{R}^m$ be such that $\lambda_I \geq 0$ and $\lambda_I^\top c_I(x_*) = 0$; let $r > 0$ and $\sigma > \|\lambda\|_D$. Show that there exists a neighborhood V of x_* in Ω such that for all $x \in V$, there holds $\ell_r(x, \lambda) \leq \Theta_\sigma(x)$.

(ii) Suppose that f and $c_{E\cup I^0_*}$ are twice differentiable at a local minimum x_* of (P_{EI}) at which the KKT conditions hold, that the semi-strong second-order sufficient condition of optimality (13.9) holds for some optimal multiplier λ_*, and that $\sigma > \|\lambda_*\|_D$. Show, using (i), that Θ_σ has a strict local minimum at x_*.

16.8. ℓ_1 *penalty function.* Suppose that f and $c_{E\cup I^0_*}$ are twice differentiable at a local minimum x_* of (P_{EI}) at which the KKT conditions hold and that the weak second-order sufficient condition of optimality (13.8) is satisfied. Positive scalars σ_i $(i \in E \cup I)$ are given and the following penalty function is considered:

$$\Theta^1_\sigma(x) = f(x) + \sum_{i\in E} \sigma_i |c_i(x)| + \sum_{i\in I} \sigma_i c_i(x)^+.$$

Show that, if $\sigma_i > |(\lambda_*)_i|$, for $i \in E \cup I$ and all optimal multiplier λ_*, then x_* is a strict local minimum of Θ^1_σ.

[*Hint*: Use the norm $v \mapsto \|v\|_P := \sum_i \sigma_i |v_i|$ and proposition 16.8.]

<u>Remark</u>: The ℓ_1-penalty function offers a natural way of controlling the magnitude of penalty parameters σ_i, when one such parameter is associated with each constraint.

16.9. *Nondifferentiable augmented Lagrangian* ([37] for equality constrained problems; [41] for an alternative to (16.22)). Suppose that f and $c_{E\cup I^0_*}$ are twice differentiable at a local minimum x_* of (P_{EI}) at which the KKT conditions hold. Let be given $\mu \in \mathbb{R}^m$ and $\sigma \in \mathbb{R}_+$. Suppose one of the following:

- (i) either the weak second-order sufficient condition of optimality (13.8) is satisfied and $\sigma > \sup\{\|\lambda_* - \mu\|_D : \lambda_* \in \Lambda_*\}$,
- (ii) or the semi-strong second-order sufficient condition of optimality (13.9) holds for some optimal multiplier λ_* and $\sigma > \|\mu - \lambda_*\|_D$.

Then $\Theta_{\mu,\sigma} : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\Theta_{\mu,\sigma}(x) := f(x) + \mu^\top c(x)^\# + \sigma \|c(x)^\#\|_P \tag{16.22}$$

has a strict local minimum at x_*.

[*Hint*: Under assumptions (i) use a technique similar to the one in the proof of proposition 16.8; under assumptions (ii) follow the same strategy as in exercise 16.7.]

17 Globalization by Line-Search

There is no guarantee that the local algorithms in chapters 14 and 15 will converge when they are started at a point x_1 far from a solution x_* to problem (P_E) or (P_{EI}). They can generate erratic sequences, which may by chance enter the neighborhood of a solution and then converge to it; but most often, the sequences will not converge. There exist several ways of damping this uncoordinated behavior and modifying the computation of the iterates so as to force their convergence. Two classes of techniques can be distinguished among them: line-search and trust-region. The former is presented in this chapter.

In methods with line-search, the iterates are generated by the recurrence

$$x_{k+1} = x_k + \alpha_k d_k,$$

where d_k is a direction in $\mathbb{R}^n$ and $\alpha_k > 0$ is a stepsize, computed by a *line-search* technique (see chapter 3), whose aim is to decrease a merit function. In this chapter, we consider algorithms in which d_k solves or approximately solves the osculating quadratic program (14.8)/(15.4) of the Newton/SQP algorithm in chapters 14/15 and the merit function is the function Θ_σ in chapter 16. For convenience, we recall the definition of Θ_σ:

$$\Theta_\sigma(x) = f(x) + \sigma \|c(x)^\#\|_{_P}, \tag{17.1}$$

where $\|\cdot\|_{_P}$ denotes an arbitrary norm and the notation $(\cdot)^\#$ was introduced on page 194. Properties of function Θ_σ are studied in chapter 16; remember that this function is usually nondifferentiable.

Let us stress the originality of this approach, which uses the solution to the osculating quadratic program to minimize Θ_σ. If d_k were an arbitrary descent direction of the nondifferentiable merit function Θ_σ, for example the steepest-descent direction, the resulting algorithm would not necessarily converge (see § 9.2.1). We shall show, however, that the difficulty coming from nonsmoothness does not occur if the search direction d_k solves the osculating quadratic problem (15.4). As for the stepsize, the value $\alpha_k = 1$ is preferred, in order to preserve the quadratic convergence of the local method. We shall see that the unit stepsize is actually accepted when x_k is close to a strong solution to (P_{EI}), provided some modifications of the search direction or the merit function are made. Therefore, the final algorithm can also be viewed as a quadratically convergent method for minimizing the structured nonsmooth function Θ_σ, a speed of convergence that cannot be obtained with general purpose nondifferentiable algorithms like those presented in part II of this book.

The concept of exactness plays an important part in the success of the approach we have just outlined. Without this property, it might indeed have been necessary to adapt σ continually to make the solution d_k to the quadratic problem a descent direction of the merit function Θ_σ. This is illustrated for an equality constraint problem in figure 17.1 (a single constraint and two

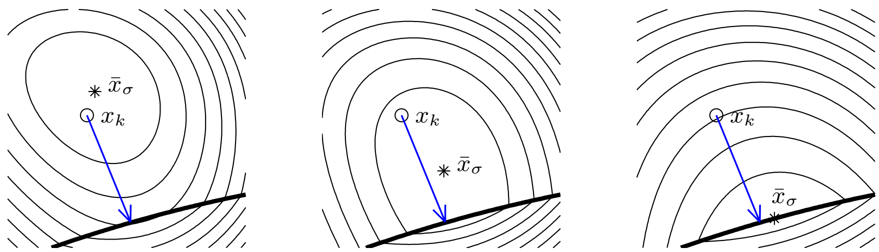

Fig. 17.1. Importance of exactness: σ too small (l), giving descent (m), giving exactness (r)

variables). The figure provides three pictures showing the level curves of Θ_σ for three increasing values of σ ($\bar{x}_\sigma$ is the minimizer of Θ_σ). They also show the constraint manifold (the bold curve at the bottom) and the Newton direction at x_k (the arrow). We assume that the current iterate x_k is close to x_* (hence the figure gives greatly enlarged views) and that the multiplier λ_k is also close to λ_*, so that the Newton direction d_k points towards x_* (this is a consequence of the quadratic convergence result in chapter 14). We can see that d_k is an ascent direction of Θ_σ if σ is not large enough (left-hand picture). In this case, there is no hope in finding a positive stepsize α_k along d_k that provides a decrease in Θ_σ. In the middle picture, σ is large enough to make d_k a descent direction of Θ_σ, although not large enough to make Θ_σ exact at x_*. In the right-hand picture, the penalty parameter σ is large enough to have $\bar{x}_\sigma = x_*$ (exactness of Θ_σ) and this gives d_k a greater chance of being a descent direction of Θ_σ. As we shall see, other conditions must also be satisfied. Observe finally that the nondifferentiability of Θ_σ manifests itself in the pictures by the lack of smoothness of its level curves when they cross the constraint manifold.

To get descent property of d_k, it will be necessary to increase σ at some iterations, but the exactness property of Θ_σ for a finite value of σ will allow the algorithm to do this finitely often. This is a very desirable property, which makes the proof of convergence possible. As soon as σ is fixed, Θ_σ

plays the role of an immutable reference, which is able to appreciate the progress towards the solution, whatever may happen to the iterates.

This chapter describes and analyzes two classes of algorithms. Line-search SQP algorithms (§ 17.1) are based on the SQP direction of chapter 15 and use line-search on Θ_σ to enforce its convergence. We derive conditions that ensure the descent property of the SQP direction on Θ_σ and study the global convergence of the algorithm. This analysis assumes the strict convexity of the osculating quadratic program defining the SQP direction (as well as its feasibility), which may require not using the Hessian of the Lagrangian, but a positive definite approximation thereof (chapter 18 explains how to generate quasi-Newton approximations). The truncated SQP algorithm of § 17.2 is presented as a line-search method that can use the exact Hessian of the Lagrangian (although we restrict the analysis to equality constrained problems). In this case, it is the way to solve the quadratic program approximately (discarding tangent negative curvature information) that allows the algorithm to generate descent directions of the merit function Θ_σ. The so-called Maratos effect (nonadmissibility of the unit stepsize asymptotically) is discussed in § 17.3, and the most common remedies for this phenomenon are described.

17.1 Line-Search SQP Algorithms

The quadratic program (QP) considered in this section is slightly more general than (15.4): the Hessian of the Lagrangian $L(x_k, \lambda_k)$ is replaced by some $n \times n$ symmetric matrix M_k. This allows us to include the Newton and the quasi-Newton versions of SQP in the same framework. On the other hand, the descent property of the QP solution and convergence of the line-search SQP algorithm often require the positive definiteness of M_k. The osculating quadratic problem in d becomes:

$$\begin{cases} \min_d \ \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c_E(x_k) + A_E(x_k) d = 0 \\ c_I(x_k) + A_I(x_k) d \le 0. \end{cases} \tag{17.2}$$

A stationary point d_k of this QP satisfies, for some multiplier $\lambda_k^{\text{QP}} \in \mathbb{R}^m$, the optimality conditions:

$$\begin{cases} (a)\ \nabla f_k + M_k d_k + A_k^\top \lambda_k^{\text{QP}} = 0 \\ (b)\ (c_k + A_k d_k)^\# = 0 \\ (c)\ (\lambda_k^{\text{QP}})_I \ge 0 \\ (d)\ (\lambda_k^{\text{QP}})_I^\top (c_k + A_k d_k)_I = 0. \end{cases} \tag{17.3}$$

For short, we have set $\nabla f_k = \nabla f(x_k)$, $c_k = c(x_k)$, and $A_k = A(x_k) = c'(x_k)$.

Let us now outline the line-search SQP algorithm that uses Θ_σ as a merit function. The description includes references to numerical techniques, whose sense will be clarified further in the section. The analysis of this algorithm is the subject of this section.

Line-search SQP:

Choose an initial iterate $(x_1, \lambda_1) \in \mathbb{R}^n \times \mathbb{R}^m$.
Compute $f(x_1)$, $c(x_1)$, $\nabla f(x_1)$, and $A_1 := c'(x_1)$.
Set $k = 1$.

1. Stop if the KKT conditions (13.1) holds at $(x_*, \lambda_*) \equiv (x_k, \lambda_k)$ (optimality is reached).
2. Compute a symmetric matrix M_k, approximating the Hessian of the Lagrangian, and find a primal-dual stationary point $(d_k, \lambda_k^{\mathrm{QP}})$ of the quadratic problem (17.2) (i.e., a solution to the optimality conditions (17.3)), which is assumed to be feasible.
3. Adapt σ_k if necessary (the update rule must satisfy (17.9) to ensure convergence, but a rule similar to the one on page 295 is often used).
4. Choose $\alpha_k > 0$ along d_k so as to obtain a "sufficient" decrease in Θ_{σ_k} (for example, use the line-search technique given on page 296).
5. Set $x_{k+1} := x_k + \alpha_k d_k$ and update $\lambda_k \to \lambda_{k+1}$.
6. Compute $\nabla f(x_{k+1})$ and $A_{k+1} := c'(x_{k+1})$.
7. Increase k by 1 and go to 1.

This algorithm does not specify how to update the dual variables λ_k. Some authors do a line-search on λ with the help of a primal-dual merit function, which therefore involves λ-values. Others compute λ_{k+1} from x_{k+1} as in the primal algorithm of § 14.3. Another possibility is also to take

$$\lambda_{k+1} := \lambda_k + \alpha_k(\lambda_k^{\mathrm{QP}} - \lambda_k), \tag{17.4}$$

where α_k is the stepsize used for the primal variables. It has already been said that the role of λ_k is less important than that of x_k, because it intervenes in the algorithm only through the matrix M_k (for example the Hessian of the Lagrangian) in (17.2). The few requirements on the way the new multiplier is determined reflects in some way this fact.

General assumptions for this section. We assume throughout this section that f and c are differentiable in an open set containing the segments $[x_k, x_{k+1}]$ that link the successive iterates. We also assume that the quadratic problem (17.2) is always feasible (i.e., its constraints are compatible).

In practice, the last assumption on the feasibility of (17.2) is far from always being satisfied at each iteration. Therefore, carefully written codes use techniques and heuristics for dealing with infeasible quadratic programs. For more computational efficiency, it is also often better to have a different penalty factor associated with each constraint, as in exercise 16.8. For simplicity, we keep a merit function with a single penalty parameter σ, knowing that an extension is possible without difficulty.

Decrease in Θ_σ Along d_k

The merit function Θ_σ decreases from x_k along d_k if d_k is a descent direction of Θ_σ at x_k (we saw in lemma 16.4 that Θ_σ has directional derivatives), meaning that

$$\Theta'_\sigma(x_k; d_k) < 0.$$

We focus on this issue in this subsection.

The next proposition identifies three conditions that make d_k a descent direction of Θ_σ: σ is large enough, M_k is positive definite, and x_k is not a stationary point of (P_{EI}). Such a result is useful for the quasi-Newton versions of SQP, where the positive definiteness of M_k is preserved. To hold, the result needs the following assumption on the norm $\|\cdot\|_P$ used in Θ_σ:

$$v \mapsto \|v^\#\|_P \text{ is convex.} \tag{17.5}$$

This hypothesis is weaker than (16.15) (see exercise 17.1).

Proposition 17.1 (descent property). *If $(d_k, \lambda_k^{\mathrm{QP}})$ satisfies the optimality conditions (17.3) and if $\|\cdot\|_P$ satisfies (17.5), then*

$$\Theta'_\sigma(x_k; d_k) \le \nabla f_k^\top d_k - \sigma\|c_k^\#\|_P = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c_k - \sigma\|c_k^\#\|_P. \tag{17.6}$$

If, in addition, $\sigma \ge \|\lambda_k^{\mathrm{QP}}\|_D$, we have

$$\Theta'_\sigma(x_k; d_k) \le -d_k^\top M_k d_k.$$

Hence $\Theta'_\sigma(x_k; d_k) < 0$, if $\sigma \ge \|\lambda_k^{\mathrm{QP}}\|_D$, if M_k is positive definite, and if x_k is not a stationary point of problem (P_{EI}).

Proof. Since a norm has directional derivatives and is Lipschitzian (like any convex function), the function $v \to \|v^\#\|_P$ has directional derivatives. From (17.5) and $(17.3)_b$, we have for $t \in\,]0,1[$:

$$\begin{aligned}
\|(c_k + tA_kd_k)^\#\|_P &= \|[(1-t)c_k + t(c_k + A_kd_k)]^\#\|_P \\
&\le (1-t)\|c_k^\#\|_P + t\|(c_k + A_kd_k)^\#\|_P \\
&= (1-t)\|c_k^\#\|_P.
\end{aligned}$$

Therefore

$$(\|\cdot^\#\|_P)'(c_k; A_kd_k) = \lim_{t\to 0+} \frac{1}{t}(\|(c_k + tA_kd_k)^\#\|_P - \|c_k^\#\|_P) \le -\|c_k^\#\|_P.$$

Then, with $(17.3)_a$, $(17.3)_b$ and $(17.3)_d$, we prove (17.6):

$$\begin{aligned}\Theta'_\sigma(x_k; d_k) &\leq \nabla f_k^\top d_k - \sigma \|c_k^\#\|_P \\ &= -d_k^\top M_k d_k - (\lambda_k^{\mathrm{QP}})^\top A_k d_k - \sigma \|c_k^\#\|_P \\ &= -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c_k - \sigma \|c_k^\#\|_P .\end{aligned}$$

If $\sigma \geq \|\lambda_k^{\mathrm{QP}}\|_D$, using $(17.3)_c$ and the generalized Cauchy-Schwarz inequality (16.14), we have

$$(\lambda_k^{\mathrm{QP}})^\top c_k - \sigma \|c_k^\#\|_P \leq (\lambda_k^{\mathrm{QP}})^\top c_k^\# - \sigma \|c_k^\#\|_P \leq (\|\lambda_k^{\mathrm{QP}}\|_D - \sigma)\|c_k^\#\|_P \leq 0.$$

Now, the second inequality of the proposition is obtained from (17.6). If $\Theta'_\sigma(x_k; d_k) = 0$ and M_k is positive definite, then $d_k = 0$. From (17.3), it follows that x_k is stationary, with λ_k^{QP} as its associated multiplier. □

Note that equality holds in (17.6) if there are only equality constraints (see the proof of lemma 17.4 below), but this is not necessarily the case when $I \neq \emptyset$ (this is the subject of exercise 17.2). Therefore, algorithms requiring the computation of $\Theta'_{\sigma_k}(x_k; d_k)$ often use the negative upper bound given by the right-hand side of (17.6):

$$\Delta_k := \nabla f_k^\top d_k - \sigma_k \|c_k^\#\|_P = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c_k - \sigma_k \|c_k^\#\|_P . \tag{17.7}$$

We have indexed σ by k, since its value will have to be modified at some iterations.

Update of the Penalty Parameter σ_k

A consequence of proposition 17.1 is that when x_k is nonstationary, when M_k is positive definite, and when σ_k satisfies

$$\sigma_k > \|\lambda_k^{\mathrm{QP}}\|_D , \tag{17.8}$$

then $\Delta_k < 0$ and the solution d_k to the osculating quadratic problem is a descent direction of Θ_{σ_k} at x_k, meaning that $\Theta'_{\sigma_k}(x_k; d_k) < 0$. Inequality (17.8) reminds us of the exactness condition $\sigma > \|\lambda_*\|_D$ found for Θ_σ in chapter 16 and is therefore natural: by maintaining (17.8) at each iteration, the algorithm ensures the exactness of Θ_σ at convergence ($\sigma_k = \sigma$ for large k and $\lambda_k^{\mathrm{QP}} \to \lambda_*$).

To maintain (17.8) at each iteration, it is necessary to modify σ_k sometimes (the evolution of λ_k^{QP} cannot be known when the algorithm is started). Global convergence will show that this inequality has to be imposed with some safeguard, given by the positive constant $\bar\sigma$ below. To keep some generality, we shall just specify the properties that an adequate adaptation rule for σ_k must enjoy:

$$\begin{cases} (a) \ \sigma_k \geq \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma}, \quad \text{for all } k \geq 1, \\ (b) \ \text{there exists an index } k_1 \text{ such that:} \\ \qquad \text{if } k \geq k_1 \text{ and } \sigma_{k-1} \geq \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma}, \text{ then } \sigma_k = \sigma_{k-1}, \\ (c) \ \text{if } \{\sigma_k\} \text{ is bounded, } \sigma_k \text{ is modified finitely often.} \end{cases} \tag{17.9}$$

Property (a) means that a little more than (17.8) must hold at each iteration. With (b), we assume that, after finitely many steps, σ_{k-1} is modified only when necessary, to obtain (a). Finally, (c) requires that each modification of σ_k is significant, so as to stabilize the sequence $\{\sigma_k\}$: asymptotically, the merit function should no longer depend on the iteration index.

It can be checked that the following rule, proposed by Mayne and Polak [250], satisfies these properties (the constant 1.5 is given to be specific; actually, any constant > 1 is appropriate):

if $\quad \sigma_{k-1} \geq \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma}$
then $\quad \sigma_k = \sigma_{k-1}$
else $\quad \sigma_k = \max(1.5\,\sigma_{k-1}, \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma})$.

Having a large parameter σ_k is harmless for the theoretical convergence, but can be disastrous in practice; so it must sometimes be decreased. In this case, the properties in (17.9) may no longer be satisfied and convergence may no longer be guaranteed. Nevertheless, an update rule like the one below is often used (the constants 1.1 and 1.5 can be replaced by any constant > 1):

UPDATE RULE FOR σ_k:
if $\quad \sigma_{k-1} \geq 1.1\,(\|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma})$,
then $\quad \sigma_k = (\sigma_{k-1} + \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma})/2$;
else $\quad$ **if** $\quad \sigma_{k-1} \geq \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma}$,
$\qquad$ **then** $\quad \sigma_k = \sigma_{k-1}$.
$\qquad$ **else** $\quad \sigma_k = \max(1.5\,\sigma_{k-1}, \|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma})$;

In this rule, when the previous penalty factor σ_{k-1} exceeds 1.1 times the minimal threshold $\|\lambda_k^{\mathrm{QP}}\|_D + \bar{\sigma}$, the new factor σ_k is set to the arithmetic mean of this threshold and of σ_{k-1}.

It is often better to use a different penalty factor for each constraint (in particular, when the constraints have very different orders of magnitude). This is done by taking as a penalty function $\Theta_\sigma(x) = f(x) + \|Sc(x)^{\#}\|_P$, where $S = \mathrm{Diag}(\sigma_1, \ldots, \sigma_n)$. The case of the ℓ_1 norm is considered in exercise 16.8.

Line-Search

The determination of the stepsize $\alpha_k > 0$ along d_k, forcing the decrease in Θ_{σ_k}, must be done in a precise manner (see § 3 for unconstrained problems). We shall enforce satisfaction of the following *Armijo condition* [12]: ω being a fixed constant in $]0, \frac{1}{2}[$, one determines $\alpha > 0$ such that

$$x_k + \alpha d_k \in \Omega \quad \text{and} \quad \Theta_{\sigma_k}(x_k + \alpha d_k) \le \Theta_{\sigma_k}(x_k) + \omega\alpha\Delta_k. \tag{17.10}$$

The requirement $\omega < \frac{1}{2}$ comes from the necessity of having asymptotic admissibility of the unit stepsize (see § 17.3); it is essential neither for consistency of (17.10) nor for global convergence ($\omega \in \,]0, 1[$ would be sufficient). The value of Δ_k in (17.10) should ideally be $\Theta'_{\sigma_k}(x_k, d_k)$, but since this directional derivative is not easy to compute, we take the negative upper bound given by (17.7).

Since $\Theta'_{\sigma_k}(x_k; d_k) \le \Delta_k < 0$ and $\omega < 1$, one can easily verify that it is possible to find $\alpha_k > 0$ satisfying (17.10). However, this Armijo condition does not eliminate unduly small α_k's, which might impair convergence of the iterates to a stationary point. This explains the following line-search algorithm. A constant $\beta \in \,]0, \frac{1}{2}]$ is chosen.

BACKTRACKING LINE-SEARCH:

Set $i = 0$, $\alpha_{k,0} = 1$.

1. If (17.10) is satisfied with $\alpha = \alpha_{k,0}$, set $\alpha_k = \alpha$ and exit.
2. Choose $\alpha_{k,i+1} \in [\beta\alpha_{k,i}, (1-\beta)\alpha_{k,i}]$.
3. Increase i by 1 and go to 1.

Taking for example $\beta = \frac{1}{2}$, the stepsize selected by this algorithm is the first element encountered in the list $\{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \cdots\}$ satisfying (17.10). Taking the first of these stepsizes does prevent α from being too small. The determination of $\alpha_{k,i+1}$ in the interval $[\beta\alpha_{k,i}, (1-\beta)\alpha_{k,i}]$ should be done using *interpolation* formulas.

Global Convergence with Positive Definite Hessian Approximations

In this subsection, we analyze the global convergence of the line-search SQP algorithm given on page 292, when σ_k is adapted by a rule satisfying properties (17.9), the stepsize α_k is determined by the line-search algorithm on page 296, and the matrices M_k used in the osculating quadratic program (17.2) are maintained positive definite, in such a way that

$$\{M_k\} \text{ and } \{M_k^{-1}\} \text{ are bounded.} \tag{17.11}$$

This is a strong assumption. For example, it is not known whether it is satisfied in the quasi-Newton versions of SQP. Besides, if $M_k = L(x_k, \lambda_k)$, positive definiteness is not guaranteed. We shall, however, accept this assumption, which allows a simple convergence proof.

Theorem 17.2 (global convergence of the line-search SQP algorithm). *Suppose that f and c are of class $C^{1,1}$ in Ω and that $\| \cdot^{\#} \|_P$ is* *convex. Consider the line-search SQP algorithm on page 292, using symmetric positive definite matrices M_k satisfying (17.11) and an update rule of σ_k satisfying (17.9). Then, starting the algorithm at a point $x_1 \in \Omega$, one of the following situations occurs:*

(*i*) *the sequence $\{\sigma_k\}$ is unbounded, in which case $\{\lambda_k^{\mathrm{QP}}\}$ is also unbounded;*
(*ii*) *there exists an index k_2 such that $\sigma_k = \sigma$ for $k \geq k_2$, and at least one of the following situations occurs:*
  (*a*) $\Theta_\sigma(x_k) \to -\infty$,
  (*b*) $\mathrm{dist}(x_k, \Omega^c) \to 0$,
  (*c*) $\nabla_x \ell(x_k, \lambda_k^{\mathrm{QP}}) \to 0$, $c_k^\# \to 0$, $(\lambda_k^{\mathrm{QP}})_I \geq 0$ *and* $(\lambda_k^{\mathrm{QP}})_I^\top (c_k)_I \to 0$.

Proof. If $\{\sigma_k\}$ is unbounded, we see from rule $(17.9)_b$ that $\{\lambda_k^{\mathrm{QP}} : \sigma_k \neq \sigma_{k-1}\}$ is unbounded. If $\{\sigma_k\}$ is bounded, rule $(17.9)_c$ shows that there exists an index k_2 such that $\sigma_k = \sigma$ for all $k \geq k_2$. It remains to show that one of the situations (*ii-a*), (*ii-b*), or (*ii-c*) occurs. For this, we suppose that (*ii-a*) and (*ii-b*) do not hold and show (*ii-c*).

Each iteration after k_2 forces the decrease in the same function Θ_σ. Since $\Theta_\sigma(x_k) \geq C > -\infty$, Armijo's condition (17.10) shows that

$$\alpha_k \Delta_k \to 0.$$

Then, if we show $\alpha_k \geq \underline{\alpha} > 0$, the result (*ii-c*) will follow. Indeed, from $\Delta_k \to 0$, (17.6) and $(17.9)_a$, we deduce

$$d_k^\top M_k d_k \to 0 \quad \text{and} \quad c_k^\# \to 0.$$

Because M_k is positive definite and has a bounded inverse, $d_k \to 0$. Then, from $(17.3)_a$ and the boundedness of M_k, we see that $\nabla_x \ell(x_k, \lambda_k^{\mathrm{QP}}) \to 0$. On the other hand, $(17.3)_c$ shows that $(\lambda_k^{\mathrm{QP}})_I \geq 0$. Finally, $\Delta_k = \nabla f_k^\top d_k - \sigma \|c_k^\#\|_P \to 0$ and $c_k^\# \to 0$ imply that $\nabla f_k^\top d_k \to 0$ and, using $(17.3)_a$, $(\lambda_k^{\mathrm{QP}})^\top A_k d_k \to 0$. Hence, from $(17.3)_d$ and $(17.3)_b$,

$$\begin{aligned}
(\lambda_k^{\mathrm{QP}})_I^\top (c_k)_I &= -(\lambda_k^{\mathrm{QP}})_I^\top (A_k d_k)_I \\
&= (\lambda_k^{\mathrm{QP}})_E^\top (A_k d_k)_E + o(1) \\
&= -(\lambda_k^{\mathrm{QP}})_E^\top (c_k)_E + o(1) \\
&= o(1),
\end{aligned}$$

because $\{\lambda_k^{\mathrm{QP}}\}$ is bounded and $(c_k)_E \to 0$.

Therefore, it remains to prove that $\alpha_k \geq \underline{\alpha} > 0$, for all k and some constant $\underline{\alpha}$. We can consider the indices k of $\mathcal{K} := \{k \geq k_2 : \alpha_k < 1\}$. Then from the rule determining the stepsize, $\alpha_k \in [\beta \bar{\alpha}_k, (1-\beta)\bar{\alpha}_k]$ for some $\bar{\alpha}_k \in \,]0, 1]$ satisfying

$$\alpha_k + \bar{\alpha}_k d_k \notin \Omega \quad \text{or} \quad \Theta_\sigma(x_k + \bar{\alpha}_k d_k) > \Theta_\sigma(x_k) + \omega \bar{\alpha}_k \Delta_k.$$

For large k, the first condition is impossible because $d_k \to 0$ would then imply that $\operatorname{dist}(x_k, \Omega^c) \to 0$. Hence, for large $k \in \mathcal{K}$, we have

$$\Theta_\sigma(x_k + \bar{\alpha}_k d_k) > \Theta_\sigma(x_k) + \omega \bar{\alpha}_k \Delta_k. \tag{17.12}$$

Let us expand the left-hand side of (17.12). Using the smoothness of f and c, $\bar{\alpha}_k \leq 1$, the convexity of $\|\cdot^\#\|_P$ (hence its Lipschitz continuity), $(17.3)_b$, and finally (17.6)–(17.7), we have successively

$$\begin{aligned}
f(x_k + \bar{\alpha}_k d_k) &= f_k + \bar{\alpha}_k \nabla f_k^\top d_k + O(\bar{\alpha}_k^2 \|d_k\|^2) \\
c(x_k + \bar{\alpha}_k d_k) &= c_k + \bar{\alpha}_k A_k d_k + O(\bar{\alpha}_k^2 \|d_k\|^2) \\
&= (1 - \bar{\alpha}_k) c_k + \bar{\alpha}_k (c_k + A_k d_k) + O(\bar{\alpha}_k^2 \|d_k\|^2) \\
\|c(x_k + \bar{\alpha}_k d_k)^\#\|_P &\leq (1 - \bar{\alpha}_k)\|c_k^\#\|_P + \bar{\alpha}_k \|(c_k + A_k d_k)^\#\|_P + O(\bar{\alpha}_k^2 \|d_k\|^2) \\
&= (1 - \bar{\alpha}_k)\|c_k^\#\|_P + O(\bar{\alpha}_k^2 \|d_k\|^2) \\
\Theta_\sigma(x_k + \bar{\alpha}_k d_k) &\leq \Theta_\sigma(x_k) + \bar{\alpha}_k \Delta_k + C_1 \bar{\alpha}_k^2 \|d_k\|^2.
\end{aligned}$$

Then (17.12) yields

$$-(1 - \omega)\bar{\alpha}_k \Delta_k \leq C_1 \bar{\alpha}_k^2 \|d_k\|^2.$$

But $\Delta_k = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c_k - \sigma \|c_k^\#\|_P \leq -d_k^\top M_k d_k \leq -C_2 \|d_k\|^2$ (boundedness of $\{M_k^{-1}\}$), so that we deduce from the above inequality:

$$\bar{\alpha}_k \geq (C_2/C_1)(1 - \omega) > 0,$$

because $\omega < 1$. The positive lower bound on α_k can therefore be taken as $\underline{\alpha} := \beta(C_2/C_1)(1 - \omega)$. This concludes the proof. □

Among the situations described in theorem 17.2, only situation (*ii-c*) is satisfactory. In this case, every cluster point of $\{(x_k, \lambda_k^{\mathrm{QP}})\}$ satisfies the optimality conditions (KKT). Unfortunately, any of the other situations may occur. For example, (*i*) may occur in the example in figure 16.1 when $\{x_k\}$ converges to x'_*, a point where λ_* is not defined. Situation (*ii-a*) will occur if, outside of the feasible set, f decreases more rapidly than $\|c(\cdot)^\#\|_P$ increases, and if x_1 is taken far enough from the feasible set; the example

$$\min\{-x^2 : x = 0\},$$

with $\|\cdot\|_P = |\cdot|$, is such. Finally, situation (*ii-b*) occurs if Ω contains no stationary point.

17.2 Truncated SQP

In this section, we consider another globalization technique of the Newton algorithm to solve the problem with only equality constraints:

$$(P_E) \quad \begin{cases} \min_x \ f(x) \\ c(x) = 0. \end{cases}$$

The local algorithm was introduced in § 14.1 and we refer the reader to § 14.4 (in the subsection entitled "The reduced system approach") for the notation. In contrast to the approach used in the previous section, we do not replace here the Hessian of the Lagrangian by a positive definite approximation. This was useful to ensure the well-posedness of the osculating quadratic program and the decrease in Θ_σ along the computed direction. Instead, we describe an algorithm that directly exploits the curvature of the problem (i.e., the second derivatives of f and c) gathered in the Hessian of the Lagrangian, even in the presence of nonconvexity.

Here also, the computed direction will be a descent direction of the merit function Θ_σ, which allows global convergence. Therefore, it must differ from Newton's direction, but the modification only needs to be done at points where the *reduced* Hessian of the Lagrangian is not positive definite. This form of weak nonconvexity can therefore be detected by the algorithm, which is a nice feature. The idea is similar to the truncated Newton algorithm in unconstrained optimization (see § 6.4): the truncated conjugate gradient (CG) algorithm is used to solve, sometimes approximately, the reduced linear system (see (14.32))

$$H_k u_k = v_k, \tag{17.13}$$

where

$$H_k := Z_k^{-\top} L_k Z_k^- \quad \text{and} \quad v_k := -g_k + Z_k^{-\top} L_k A_k^- c_k. \tag{17.14}$$

Note that the reduced Hessian of the Lagrangian H_k is symmetric but may be indefinite. By the truncated CG, the algorithm aims at collecting only the "positive definite part" of H_k. This is obtained by stopping the CG iterations certainly before a conjugate direction w is a negative curvature direction for H_k (more precisely, before $w^\top H_k w$ becomes less than an appropriate positive threshold). Let us denote by $\tilde{u}_k$ the approximate solution to (17.13) computed by the truncated CG algorithm. We shall show that the search direction

$$d_k = -A_k^- c_k + Z_k^- \tilde{u}_k \tag{17.15}$$

is then a descent direction of Θ_σ provided σ is larger than an easily computable threshold. Another interesting property of this approach is that, since H_k is positive definite around a strong solution to (P_E), the CG iterations can be pursued up to completion close to such a solution, so that local quadratic convergence is not prevented.

Let us look at this in more detail.

Truncated CG Iterations

The *truncated conjugate gradient* (TCG) *algorithm* to solve (17.13) is presented below. For clarity, we drop the index k of the Newton algorithm and denote by i the CG iteration index (in superscript). For $i = 0, \ldots, j$, Algorithm TCG generates iterates u^i, approximating the solution to (17.13), residuals $r^i := Hu^i - v$, and conjugate directions w^i. The algorithm can be stopped at any iteration (global convergence of the truncated SQP method will not be affected by this), but it must certainly be interrupted at u^j if the next conjugate direction w^j is a *quasi-negative curvature* direction for H. This means that the following inequality does not hold with $i = j$:

$$(w^i)^\top H w^i \geq \nu \|w^i\|_2^2. \tag{17.16}$$

The threshold $\nu > 0$ is assumed to be independent of the index k, although an actual implementation would use a more sophisticated rule for setting this parameter, allowing small values when approaching a solution. Hence, Algorithm TCG simply discards quasi-negative directions. It is in this way that nonconvexity is dealt with.

ALGORITHM TCG FOR (17.13):

1. Choose $\nu > 0$. Set $u^0 = 0$ and $r^0 = -v$, where v is defined by (17.14).
2. For $i = 0, 1, \ldots$ do the following:
   2.1. If desired or if $r^i = 0$, stop to iterate and go to step 3 with $j = i$.
   2.2. Compute a new conjugate direction:
   $$w^i = \begin{cases} -r^i & \text{if } i = 0 \\ -r^i + \frac{\|r^i\|^2}{\|r^{i-1}\|^2}\, w^{i-1} & \text{if } i \geq 1. \end{cases}$$
   2.3. Compute $p^i = Hw^i$.
   2.4. If (17.16) does not hold, go to step 3 with $j = i$.
   2.5. Compute the new iterate $u^{i+1} = u^i + t^i w^i$ and the new residual $r^{i+1} = r^i + t^i p^i$, with the stepsize
   $$t^i = \frac{\|r^i\|^2}{(w^i)^\top p^i}.$$
3. Take as the approximate solution to (17.13):
$$\tilde{u} = \begin{cases} v & \text{if } j = 0 \\ u^j & \text{if } j \geq 1. \end{cases}$$

Observe that, since the first iterate of Algorithm TCG is $u^0 = 0$, the first CG direction is $w^0 = -r^0 = v$, the right-hand side of (17.13). This is important for the analysis that follows. Another key point is that the directions w^i are conjugate: $w^{i_1} H w^{i_2} = 0$ for $i_1 \neq i_2$. Note finally that Algorithm TCG chooses to output the approximate solution u^j currently obtained when $j \geq 1$ (it is different from zero), but $\tilde{u} = w^0 = v$ when $j = 0$ ($u^0 = 0$ in this case).

Lemma 17.3. *The vector $\tilde{u}$ computed by Algorithm TCG has the form*

$$\tilde{u} = Jv, \tag{17.17}$$

where J is the identity matrix when $j = 0$ and

$$J = \sum_{i=0}^{j-1} \frac{w^i (w^i)^\top}{(w^i)^\top H w^i} \tag{17.18}$$

when $j \geq 1$. Furthermore $\|J\|_2 \leq \max\left(1, \frac{j}{\nu}\right)$.

Proof. If $i = 0$, $u = v$ and the result follows. Otherwise Algorithm TCG generates conjugate directions $w^0, \ldots, w^{j-1}$. By orthogonality of r^i and w^{i-1}, by the fact that the algorithm starts with $u^0 = 0$, and by conjugacy of the directions w^i, one has for $1 \leq i \leq j$:

$$\begin{aligned}
\|r^i\|^2 &= -(w^i)^\top r^i \\
&= -(w^i)^\top (Hu^i - v) \\
&= -(w^i)^\top H \left(\sum_{l=0}^{i-1} t^l w^l \right) + (w^i)^\top v \\
&= (w^i)^\top v.
\end{aligned}$$

Also, $\|r^0\|^2 = (w^0)^\top v$. Therefore

$$\tilde{u} = \sum_{i=0}^{j-1} t^i w^i = \sum_{i=0}^{j-1} \frac{(w^i)^\top v}{(w^i)^\top H w^i} \, w^i = \left(\sum_{i=0}^{j-1} \frac{w^i (w^i)^\top}{(w^i)^\top H w^i} \right) v.$$

This proves (17.18).

The upper bound on $\|J\|_2$ comes from the fact that $\|vv^\top\|_2 = \|v\|_2^2$ and (17.16). $\square$

Note that, when $j \geq 1$, the matrix J is positive semi-definite with rank j. In view of (17.13) and (17.17), this matrix appears as a kind of "pseudo-inverse of the positive definite part" of H.

Descent Property

In the next lemma, we give conditions ensuring that the direction d_k given by (17.15) is a descent direction of Θ_{σ_k}. For this, it is convenient to give another expression of d_k by introducing the following right inverse of A_k:

$$\widetilde{A}_k^- := (I - Z_k^- J_k Z_k^{-\top} L_k) A_k^-. \tag{17.19}$$

This is the right inverse $\widehat{A}_k^-$ in (14.33), in which H_k^{-1} has been substituted by its approximation J_k. Then

$$d_k = \tilde{r}_k + \tilde{t}_k, \tag{17.20}$$

where

$$\tilde{r}_k = -\widetilde{A}_k^- c_k \quad \text{and} \quad \tilde{t}_k = -Z_k^- J_k g_k.$$

We also use the multiplier associated with $\widetilde{A}_k^-$:

$$\tilde{\lambda}_k = -\widetilde{A}_k^{-\top} \nabla f_k. \tag{17.21}$$

How to compute this multiplier efficiently is dealt with in the next subsection.

Lemma 17.4 (descent property). *Suppose that f and c are differentiable at x_k. Let d_k be given by* (17.15)*, where $\tilde{u}_k$ is the approximate solution to* (17.13) *computed by Algorithm TCG. Then Θ_{σ_k} has a directional derivative in the direction d_k, whose value is given by*

$$\Theta'_{\sigma_k}(x_k; d_k) = -g_k^\top J g_k + \tilde{\lambda}_k^\top c_k - \sigma_k \|c_k\|_P. \tag{17.22}$$

It is negative if x_k is nonstationary and $\sigma_k > \|\tilde{\lambda}_k\|_D$.

Proof. Since a norm is Lipschitz continuous and has directional derivatives, $\|\cdot\|_P \circ c$ has directional derivatives (see lemma 13.1). Using the fact that d_k satisfies the linearized constraints (i.e., $A_k d_k = -c_k$), one has $(\|\cdot\|_P \circ c)'(x_k; d_k) = (\|\cdot\|_P)'(c_k; -c_k) = -\|c_k\|_P$. Therefore

$$\Theta'_{\sigma_k}(x_k; d_k) = \nabla f_k^\top d_k - \sigma_k \|c_k\|_P.$$

Using (17.20) and (17.21), we get (17.22).

Suppose now that $\sigma_k > \|\tilde{\lambda}_k\|_D$. Since $\tilde{\lambda}_k^\top c_k \le \|\tilde{\lambda}_k\|_D \|c_k\|_P$, we obtain

$$\Theta'_{\sigma_k}(x_k; d_k) \le -g_k^\top J_k g_k + (\|\tilde{\lambda}_k\|_D - \sigma_k)\|c_k\|_P \le 0.$$

If $\Theta'_{\sigma_k}(x_k; d_k) = 0$, it follows that $c_k = 0$ and $g_k^\top J_k g_k = 0$. If the number of CG iterations $j_k = 0$, then $J_k = I$, hence $g_k = 0$ and x_k is stationary. It remains to show that j_k cannot be ≥ 1 when $\Theta'_{\sigma_k}(x_k; d_k) = 0$. If $j_k \ge 1$, one would have $v_k \ne 0$ (see step 2.1 of Algorithm TCG) and therefore $g_k \ne 0$ (since $c_k = 0$). But with the structure of J_k and the fact that $w_k^0 = v_k = -g_k$ when $c_k = 0$, one would have $g_k^\top J_k g_k \ge (g_k^\top w_k^0)^2 / ((w_k^0)^\top H_k w_k^0) = \|g_k\|^4 / (g_k^\top H g_k) > 0$, which would contradict the fact that $g_k^\top J_k g_k = 0$. $\square$

Computation of $\tilde{\lambda}_k$

Let us drop the index k. From (17.21) and (17.19), the definition of $\tilde{\lambda}$ involves the matrix J:

$$\tilde{\lambda} = -A^{-\top}(\nabla f - LZ^{-}Jg).$$

We do not want to store this matrix, however. In fact, to compute $\tilde{\lambda}$, one has to evaluate $\bar{u} = Jg$, which is the approximate solution to

$$H\bar{u} = g, \tag{17.23}$$

obtained by using the same conjugate directions w^i and the same products $p^i = Hw^i$, $i = 0, \dots, j-1$, as those used to compute the approximate solution $\tilde{u}$ to (17.13) by Algorithm TCG. The computation of $\tilde{u}$ and $\bar{u}$ can be made in parallel, hence avoiding the need to store the conjugate directions w^i (or J) or the need to compute twice the Hessian-vector products $p^i = Hw^i$. This is what Algorithm TCG2 below does. Its outputs are $\tilde{u}$ and $\bar{u}$.

ALGORITHM TCG2 FOR (17.13) AND (17.23):

1. Choose $\nu > 0$. Set $u^0 = 0$, $r^0 = -v$, $\bar{u}^0 = 0$, and $\bar{r}^0 = -g$, where v is defined by (17.14).
2. For $i = 0, 1, \dots$ do the following:
   2.1. If desired or if $r^i = 0$, stop to iterate and go to step 3 with $j = i$.
   2.2. Compute a new conjugate direction:
   $$w^i = \begin{cases} -r^i & \text{if } i = 0 \\ -r^i + \frac{\|r^i\|^2}{\|r^{i-1}\|^2}\, w^{i-1} & \text{if } i \geq 1. \end{cases}$$
   2.3. Compute $p^i = Hw^i$.
   2.4. If (17.16) does not hold, go to step 3 with $j = i$.
   2.5. Compute the new iterates $u^{i+1} = u^i + t^i w^i$ and $\bar{u}^{i+1} = \bar{u}^i + \bar{t}^i w^i$ and the new residuals $r^{i+1} = r^i + t^i p^i$ and $\bar{r}^{i+1} = \bar{r}^i + \bar{t}^i p^i$, with the stepsizes
   $$t^i = \frac{\|r^i\|^2}{(w^i)^\top p^i} \quad \text{and} \quad \bar{t}^i = -\frac{(\bar{r}^i)^\top w^i}{(w^i)^\top p^i}.$$
3. Take as the approximate solution to (17.13) and (17.23):
$$\tilde{u} = \begin{cases} v & \text{if } j = 0 \\ w^j & \text{if } j \geq 1 \end{cases} \quad \text{and} \quad \bar{u} = \begin{cases} g & \text{if } j = 0 \\ \bar{u}^j & \text{if } j \geq 1. \end{cases}$$

It may occur that the linear system (17.23) is solved before (17.13). In this case, the stepsizes $\bar{t}^i$ vanish and $\bar{u}^i$ is no longer modified. It is easy to verify that $\tilde{\lambda}$ is obtained from $\bar{u}$ by:

$$\tilde{\lambda} = -A^{-\top}(\nabla f - LZ^{-}\bar{u}). \tag{17.24}$$

Indeed, since $\bar{u}^0 = 0$, one has for $1 \leq i \leq j$:

$$(w^i)^\top \bar{r}^i = (w^i)^\top (H\bar{u}^i - g) = (w^i)^\top H \left(\sum_{l=0}^{i-1} \bar{t}^l w^l \right) - (w^i)^\top g = -(w^i)^\top g.$$

Hence

$$\bar{u} = \sum_{i=0}^{j-1} \bar{t}^i w^i = \sum_{i=0}^{j-1} \frac{(w^i)^\top g}{(w^i)^\top H w^i} w^i = Jg.$$

The Truncated SQP Algorithm and its Global Convergence

The truncated SQP algorithm to solve problem (P_E) generates a sequence $\{x_k\}_{k\geq 1}$ by the recurrence

$$x_{k+1} = x_k + \alpha_k d_k,$$

where the direction $d_k \in \mathbb{R}^n$ is determined by (17.15), with $\tilde{u}_k$ computed by Algorithm TCG2, and the stepsize $\alpha_k > 0$ is determined by a line-search on the merit function Θ_{σ_k}.

According to lemma 17.4, d_k is a descent direction of Θ_{σ_k} provided x_k is nonstationary and $\sigma_k > \|\tilde{\lambda}_k\|_D$. This requires a modification of σ_k at some iterations and we assume that a rule respecting conditions similar to (17.9) is adopted: for some fixed constant $\bar{\sigma} > 0$, the following holds

$$\begin{cases} (a)\ \sigma_k \geq \|\tilde{\lambda}_k\|_D + \bar{\sigma}, \quad \text{for all } k \geq 1, \\ (b)\ \text{there exists an index } k_1 \text{ such that:} \\ \quad\ \text{if } k \geq k_1 \text{ and } \sigma_{k-1} \geq \|\tilde{\lambda}_k\|_D + \bar{\sigma}, \text{ then } \sigma_k = \sigma_{k-1}, \\ (c)\ \text{if } \{\sigma_k\} \text{ is bounded, } \sigma_k \text{ is modified finitely often.} \end{cases} \tag{17.25}$$

Since at a nonstationary iterate x_k, d_k is a descent direction of Θ_{σ_k}, one can determine a stepsize $\alpha_k > 0$ such that the following Armijo inequality holds

$$\Theta_{\sigma_k}(x_k + \alpha_k d_k) \leq \Theta_{\sigma_k}(x_k) + \omega \alpha_k \Theta'_{\sigma_k}(x_k; d_k), \tag{17.26}$$

where ω is a constant chosen in $]0, \frac{1}{2}[$. As in the line-search SQP algorithm on page 292, the stepsize is determined in step 4 below by *backtracking*.

We can now summarize the overall TSQP algorithm to solve the equality constrained problem (P_E).

ALGORITHM TSQP:

Choose an initial iterate $(x_1, \lambda_1) \in \mathbb{R}^n \times \mathbb{R}^m$.
Compute $f(x_1)$, $c(x_1)$, $\nabla f(x_1)$, and $A(x_1)$.
Set the constants $\nu > 0$ (quasi-negative curvature threshold), $\omega \in]0, \frac{1}{2}[$ (slope modifier in the Armijo condition), $\bar{\sigma} > 0$ (penalty parameter threshold), and $\beta \in]0, \frac{1}{2}]$ (backtracking safeguard parameter).
Set $k = 1$.

1. *Stopping test*: Stop if $c_k = 0$ and $g_k = 0$.
2. *Step computation*:
   - Compute the restoration step $r_k = -A_k^- c_k$.
   - Compute the reduced gradient $g_k = Z_k^{-\top} \nabla f_k$ and the right-hand side of (17.13) $v_k = -g_k - Z_k^{-\top} L_k r_k$.
   - Run Algorithm TCG2 to compute $\tilde{u}_k$ and $\bar{u}_k$.
   - Compute the full step $d_k = r_k + Z_k^- \tilde{u}_k$ and the multiplier $\tilde{\lambda}_k$ by (17.24).
3. *Penalty parameter setting*: Update σ_k such that (17.25) holds.
4. *Backtracking line-search*:
   - Set $\alpha = 1$.
   - While α does not satisfy Armijo's inequality (17.26), pick a new stepsize α in $[\beta\alpha, (1-\beta)\alpha]$.
   - Set $\alpha_k = \alpha$.
5. *New iterates*: Set $x_{k+1} = x_k + \alpha_k d_k$ and $\lambda_{k+1} = \lambda_{k+1}^{\text{LS}}$.
6. Increase k by 1 and go to 1.

Before proving the global convergence of this algorithm, let us make some observations. In a practical algorithm, the stopping test in step 1 would be replaced by a condition checking that c_k and g_k are sufficiently small. In practice, in step 4, the new stepsize chosen in the interval $[\beta\alpha, (1-\beta)\alpha]$ during the line-search should be obtained by *interpolation*. In step 5, we have set the new multiplier λ_{k+1} to the least-squares multiplier

$$\lambda_k^{\text{LS}} := -A_k^{-\top} \nabla f_k.$$

This makes Algorithm TSQP close to the primal version of Newton's algorithm analyzed in theorem 14.5. Another possibility would have been to choose $\lambda_{k+1} = \tilde{\lambda}_k$. Observe however that, even if the CG iterations of Algorithm TCG2 solve (17.13) and (17.23) exactly, $\tilde{\lambda}_k \neq \lambda_k^{\text{QP}}$ (in this case $\tilde{\lambda}_k = \widehat{\lambda}_k$ given by (14.36), compare with (14.35)), so that with that choice of λ_{k+1}, Algorithm TSQP does not reduce to Newton's algorithm in a neighborhood of a strong solution.

Theorem 17.5 (global convergence of the line-search truncated SQP algorithm). *Suppose that the functions f and c are twice continuously differentiable with Lipschitz continuous first derivatives. Suppose also that the sequences $\{\nabla f_k\}$, $\{L_k\}$, $\{A_k^-\}$, and $\{Z_k^-\}$ generated by Algorithm TSQP are bounded. Then the sequence of penalty parameters $\{\sigma_k\}$ is stationary for sufficiently large k: $\sigma_k = \sigma$. If furthermore $\{\Theta_\sigma(x_k)\}$ is bounded below, the sequences $\{c_k\}$ and $\{g_k\}$ converge to 0.*

Proof. We denote by $C_1, C_2, \ldots$ positive constants, independent of k. We can assume that $\|c_k\| + \|g_k\| > 0$ for all $k \geq 1$, because otherwise the conclusion is clear.

Note first, that the assumptions imply the boundedness of $\{\tilde{\lambda}_k\}$ (use (17.24), the boundedness of $\{A_k^-\}$, $\{\nabla f_k\}$, $\{L_k\}$, $\{Z_k^-\}$, and that of $\{J_k\}$ given by lemma 17.3). Then by $(17.25)_b$, $\{\sigma_k\}$ is also bounded, hence stationary for large enough k (use $(17.25)_c$). From Armijo's inequality (17.26), $\Theta_\sigma(x_k)$ is decreasing. It is also bounded below (by assumption), hence it converges. This implies that $\alpha_k \Theta'_\sigma(x_k; d_k)$ tends to 0, or equivalently (use lemma 17.4 and $(17.25)_a$)

$$\alpha_k g_k^\top J_k g_k \to 0 \quad \text{and} \quad \alpha_k c_k \to 0. \tag{17.27}$$

Let us now show that $\{\alpha_k\}$ is bounded away from 0. From the line-search (step 4), when $\alpha_k < 1$, there is a stepsize $\overline{\alpha}_k \in \,]0,1]$ such that $\alpha_k \in [\beta\overline{\alpha}_k, (1-\beta)\overline{\alpha}_k]$ and

$$\Theta_\sigma(x_k + \overline{\alpha}_k d_k) > \Theta_\sigma(x_k) + \omega\overline{\alpha}_k \Theta'_\sigma(x_k; d_k).$$

Using the smoothness of f and c and the fact that d_k satisfies the linearized constraints, one has successively

$$\begin{aligned}
f(x_k + \overline{\alpha}_k d_k) &= f(x_k) + \overline{\alpha}_k f'(x_k) \cdot d_k + O(\overline{\alpha}_k^2 \|d_k\|^2), \\
c(x_k + \overline{\alpha}_k d_k) &= (1 - \overline{\alpha}_k) c(x_k) + O(\overline{\alpha}_k^2 \|d_k\|^2), \\
\Theta_\sigma(x_k + \overline{\alpha}_k d_k) &\leq \Theta_\sigma(x_k) + \overline{\alpha}_k \Theta'_\sigma(x_k; d_k) + C_1 \overline{\alpha}_k{}^2 \|d_k\|^2.
\end{aligned}$$

Therefore $(\omega - 1)\Theta'_\sigma(x_k; d_k) < C_1 \overline{\alpha}_k \|d_k\|^2$ or

$$g_k^\top J_k g_k + \|c_k\|_P < C_2 \overline{\alpha}_k \|d_k\|^2, \tag{17.28}$$

where $C_2 = C_1/((1-\omega)\min(1, \bar{\sigma}))$. With the boundedness of $\{A_k^-\}$, $\{Z_k^-\}$, $\{L_k\}$, and $\{J_k\}$, we have $d_k = O(\|J_k^{1/2} v_k\| + \|c_k\|_P)$ and, due to the form of v_k, $d_k = O(\|J_k^{1/2} g_k\| + \|c_k\|_P)$. Then, inequality (17.28) becomes

$$g_k^\top J_k g_k + \|c_k\|_P < C_3 \overline{\alpha}_k (g_k^\top J_k g_k + \|c_k\|_P^2).$$

From (17.27), $\alpha_k c_k \to 0$ and therefore for large k

$$g_k^\top J_k g_k < C_3 \overline{\alpha}_k g_k^\top J_k g_k.$$

This inequality shows that $g_k^\top J_k g_k \neq 0$ when $\alpha_k < 1$ and k is large enough and that $\{\overline{\alpha}_k\}$ is bounded away from zero. Since $\alpha_k \geq \beta\overline{\alpha}_k$, $\{\alpha_k\}$ is also bounded away from zero.

From (17.27)

$$g_k^\top J_k g_k \to 0 \quad \text{and} \quad c_k \to 0. \tag{17.29}$$

It remains to show that $g_k \to 0$. Assume the opposite: there is a constant $\gamma > 0$ and subsequence $\mathcal{K}$ such that $\|g_k\| \geq \gamma$ for $k \in \mathcal{K}$. Using the first term of the expression (17.18) of J_k when $j_k \geq 1$, $w_k^0 = v_k$, and the boundedness of $\{H_k\}$, one can write

$$g_k^\top J_k g_k \geq \min\left(\|g_k\|_2^2, \frac{(g_k^\top v_k)^2}{v_k^\top H_k v_k}\right) \geq \min\left(\gamma^2, C_4 \frac{(g_k^\top v_k)^2}{\|v_k\|^2}\right).$$

The numerator can be bounded below as follows:

$$\begin{aligned}
(g_k^\top v_k)^2 &= [-\|g_k\|^2 + O(\|g_k\|\,\|c_k\|)]^2 \\
&= \|g_k\|^4 + O(\|g_k\|^3\,\|c_k\|) + O(\|g_k\|^2\,\|c_k\|^2) \\
&\geq \frac{1}{2}\|g_k\|^4 - C_5\|g_k\|^2\,\|c_k\|^2 \\
&\geq \|g_k\|^2(\frac{1}{2}\gamma^2 - C_5\|c_k\|^2),
\end{aligned}$$

which is positive for large k in $\mathcal{K}$. For the denominator, we use the upper bound:

$$\|v_k\|^2 \leq 2\|g_k\|^2 + C_6\|c_k\|^2 \leq \|g_k\|^2(2 + C_6\|c_k\|^2/\gamma^2).$$

Therefore for large k in $\mathcal{K}$:

$$g_k^\top J_k g_k \geq \min\left(\gamma^2, \frac{\frac{1}{2}\gamma^2 - C_5\|c_k\|^2}{2 + C_6\|c_k\|^2/\gamma^2}\right).$$

This is in contradiction with (17.29). □

17.3 From Global to Local

In this section, we analyze conditions under which the line-search algorithms of the present chapter can transform themselves into the “local” algorithms of chapter 14. In view of the quadratic convergence of the local methods, this “mutation” is highly desirable. Because the direction generated by the local algorithm is used as a descent direction of some merit function, this transformation will occur if the line-search accepts the unit stepsize during the last iterations. This property is referred to as the *asymptotic admissibility of the unit stepsize*. We shall see that it is not guaranteed without certain modifications of the algorithms, which are therefore crucial for their efficiency.

For simplicity, we assume in this section that the problem has only equality constraints:

$$(P_E) \quad \begin{cases} \min_x \ f(x) \\ c(x) = 0. \end{cases}$$

Since our study is asymptotic, assuming convergence of the sequence $\{(x_k, \lambda_k)\}$ to a primal-dual solution (x_*, λ_*), this simplification amounts to assuming that the active constraints are identified after finitely many iterations, in which case problem (P_{EI}) reduces locally to a problem with only equality constraints (theorem 15.2 tells us something about this).

The Maratos Effect

The merit function Θ_σ introduced in § 16.4 and defined by

$$\Theta_\sigma(x) = f(x) + \sigma \|c(x)\|_P$$

does not necessarily accept unit stepsizes asymptotically. This is known as the *Maratos effect*. We mean by this that when d_k solves the quadratic problem

$$\begin{cases} \min_d \ \nabla f(x_k)^\top d + \frac{1}{2} d^\top M_k d \\ c(x_k) + A(x_k) d = 0, \end{cases} \tag{17.30}$$

we may have

$$\Theta_\sigma(x_k + d_k) > \Theta_\sigma(x_k), \tag{17.31}$$

however close to (x_*, L_*) the current pair (x_k, M_k) may be.

The following counter-example demonstrates this fact. There, the considered iterate x_k is on the constraint manifold: $c(x_k) = 0$. We have seen in proposition 17.1 that, if $\sigma_k \geq \|\lambda_k^{\mathrm{QP}}\|_D$ and M_k is positive definite, Θ_{σ_k} decreases along the Newton direction d_k, which means that, for small stepsizes, the decrease in f along d_k compensates the increase in $\|c\|_P$. In the counter-example, this compensation not longer holds for stepsizes close to 1.

Counter-example 17.6. Consider the problem on $\mathbb{R}^2$

$$\begin{cases} \min_x \ -x_1 + \tau(x_1^2 + x_2^2 - 1) \\ x_1^2 + x_2^2 - 1 = 0, \end{cases}$$

where $\tau \in \mathbb{R}$. Its unique solution is $x_* = (1, 0)$ and the associated multiplier is $\lambda_* = \frac{1}{2} - \tau$. The Hessian of the Lagrangian at the solution is $L_* = I$.

Suppose now that the step d at x is given by the osculating quadratic problem, defined at a feasible point x with the matrix $M = L_* = I$:

$$\begin{cases} \min_d \ -d_1 + \frac{1}{2}\|d\|_2^2 \\ x^\top d = 0. \end{cases}$$

Its solution for $x = (\cos\theta, \sin\theta)$ lying on the constraint is

$$d = \begin{pmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{pmatrix}$$

and $c(x+\alpha d) = \alpha^2 \sin^2\theta$. Hence, if $\|\cdot\|_P = |\cdot|$,

$$\begin{aligned} \Theta_\sigma(x) &= -\cos\theta \\ \Theta_\sigma(x+\alpha d) &= -\cos\theta - \alpha\sin^2\theta + (\tau+\sigma)\alpha^2\sin^2\theta. \end{aligned}$$

Then $\Theta_\sigma(x+d) > \Theta_\sigma(x)$ whenever $\tau+\sigma > 1$ (and $\theta \neq 0$). Because $\sigma \geq |\lambda_*| \equiv |\frac{1}{2}-\tau|$ is needed to have an exact penalty, Θ_σ increases for a unit stepsize if $\tau > \frac{3}{4}$.

Figure 17.2 shows the level curves of Θ_σ around the solution for $\tau = 1$ and $\sigma = 0.6$, as well as the Newton step d from an x on the constraint manifold (the bold curve), rather close to the solution $(1,0)$. One clearly observes that $\Theta_\sigma(x+d) > \Theta_\sigma(x)$. □

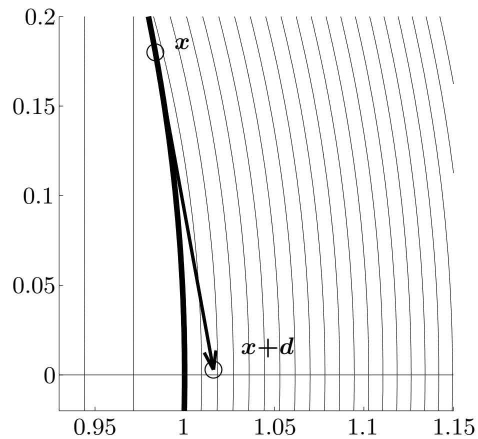

Fig. 17.2. Example with a Maratos effect

This phenomenon somehow reveals a discrepancy between Θ_σ and the osculating quadratic problem used to compute d_k. Since this model is good (it yields local quadratic convergence), the blame must be put on the merit function, or on the way in which it is used. In the rest of this section, we analyze different remedies for the Maratos effect and prove that they are effective close to a solution. The Maratos effect can also occur far from a solution and it is then more difficult to deal with. The first remedy consists in modifying the step d_k by adding to it a small displacement, called a *second order correction*, that does not prevent quadratic convergence. Another possibility is to modify the merit function, which is considered next.

Modification of the Step: Second Order Correction

Example 17.6 has shown that there are situations in which, even close to the solution, the increase in $\|c(\cdot)\|_P$ from x_k to $x_k + d_k$ is not compensated by a decrease in f, resulting finally in an increase in Θ_σ. The remedy for the Maratos effect presented in this subsection consists in adding to d_k a small correcting step $e_k \in \mathbb{R}^n$, whose aim is to decrease $\|c(\cdot)\|_P$. This additional step is defined by

$$e_k = -A_k^- c(x_k+d_k), \tag{17.32}$$

where A_k^- is some right inverse of the Jacobian matrix $A_k = c'(x_k)$, which is assumed to be surjective. Hence, e_k is a constraint-restoration step at x_k+d_k. Figure 17.3 shows the second order correction for counter-example 17.6: the small step e from $x+d$ to $x+d+e$.

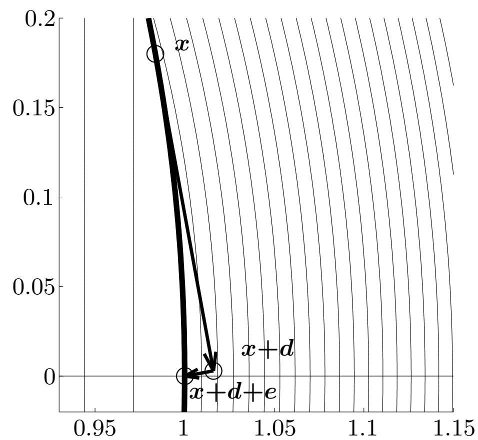

Fig. 17.3. Second order correction

One speaks of *second-order correction* because $c(x_k+d_k) = O(\|d_k\|^2)$ and therefore $e_k = O(\|d_k\|^2)$ is of order 2 in d_k. This modification of d_k preserves a possible quadratic convergence since, assuming $x_k+d_k-x_* = O(\|x_k-x_*\|^2)$, we have

$$x_k + d_k + e_k - x_* = (x_k + d_k - x_*) + e_k = O(\|x_k - x_*\|^2),$$

owing to the preceding estimate of e_k and to the fact that $d_k \sim (x_k - x_*)$ (lemma 13.5).

Because e_k is computed by evaluating c at a point different from x_k, it cannot be guaranteed that $d_k + e_k$ is a descent direction of Θ_{σ_k} at x_k. Therefore, a line-search along this direction may be impossible. The least expensive approach is then to determine a stepsize $\alpha_k > 0$ along the arc

$$\alpha \mapsto p_k(\alpha) = x_k + \alpha d_k + \alpha^2 e_k.$$

It has the descent direction d_k as a tangent at $\alpha = 0$ and visits $x_k + d_k + e_k$ for $\alpha = 1$. The stepsize α_k can be computed in the same way as along d_k, forcing at each iteration the inequality

$$\Theta_{\sigma_k}(x_k+\alpha_k d_k+\alpha_k^2 e_k) \leq \Theta_{\sigma_k}(x_k) + \omega\alpha_k\Theta'_{\sigma_k}(x_k; d_k), \tag{17.33}$$

for some $\alpha_k \in]0, 1]$. It is easy to verify that this inequality can always be satisfied, provided d_k is a descent direction of Θ_{σ_k} at x_k.

In the next proposition, we give conditions under which the unit stepsize $\alpha_k = 1$ is accepted in (17.33) when x_k is near a strong solution to (P_E). Part of these conditions is related to the matrix M_k, which must satisfy (17.34). This condition is of the form $t_k \geq o(\|d_k\|^2)$, for some real numbers t_k, which means that there must exist a sequence of real numbers $\{s_k\}$, such that $t_k \geq s_k$ and $s_k = o(\|d_k\|^2)$ when $k \to \infty$. Observe that this condition is satisfied when M_k is "large enough". This is not surprising, since then the tangent step is small (see remark 2 on page 235) and the total step d_k is close to the restoration step, along which the unit stepsize is known to be accepted by the norm of the constraints (see exercise 17.4). Observe also that condition (17.34) is satisfied when M_k is the Hessian of the Lagrangian (with convergent multipliers), which corresponds to Newton's method.

Proposition 17.7 (admissibility of the unit step-size with a second order correction). *Suppose that f and c are of class C^2 in a neighborhood of a solution x_* to (P_E) satisfying the second-order sufficient conditions of optimality and at which $A_* = c'(x_*)$ is surjective. Let $\{x_k\}$ be a sequence converging to x_*, let d_k satisfy the first-order optimality conditions of the osculating quadratic problem* (17.30)*, and let e_k be defined by* (17.32)*. Suppose also that*

- *$\{A_k^-\}$ is bounded and $d_k \to 0$,*
- *the matrix M_k used in the osculating quadratic problem* (17.30) *over-estimates the Hessian of the augmented Lagrangian $L_*^r := L_* + rA_*^\top A_*$, in the sense that*
$$d_k^\top(M_k - L_*^r)d_k \geq o(\|d_k\|^2), \tag{17.34}$$
where $r \geq 0$ is such that L_*^r is positive definite (such an r always exists under the assumptions already stated, see exercise 16.1*),*
- *the penalty parameter σ_k used in Θ_{σ_k} satisfies*
$$\|\lambda_k^{\mathrm{QP}}\|_D \leq \sigma_k \leq \hat{\sigma}, \tag{17.35}$$
where λ_k^{QP} is a multiplier associated with the constraints of (17.30) *and $\hat{\sigma}$ is a constant.*

Then, for $\omega < \frac{1}{2}$ and large enough k, there holds

$$\Theta_{\sigma_k}(x_k+d_k+e_k) \leq \Theta_{\sigma_k}(x_k) + \omega\Theta'_{\sigma_k}(x_k; d_k).$$

Proof. Despite the nondifferentiability of Θ_{σ_k}, one can obtain an expansion of $\Theta_{\sigma_k}(x_k{+}d_k{+}e_k)$ with a precision of order $o(\|d_k\|^2)$. This one follows from an expansion of $f(x_k{+}d_k{+}e_k)$ and $c(x_k{+}d_k{+}e_k)$ about x_k. Using the smoothness assumptions on f and c, the constraint in (17.30), the definition of e_k in (17.32), the boundedness of $\{A_k^-\}$, and the optimality of (x_*, λ_*), we have successively

$$\begin{aligned}
c(x_k{+}d_k) &= c_k + A_k d_k + \frac{1}{2}c''(x_*)\cdot d_k^2 + o(\|d_k\|^2),\\
&= \frac{1}{2}c''(x_*)\cdot d_k^2 + o(\|d_k\|^2),\\
e_k &= O(\|c(x_k{+}d_k)\|)\\
&= O(\|d_k\|^2),\\
c(x_k{+}d_k{+}e_k) &= c(x_k{+}d_k) + A_k e_k + o(\|e_k\|)\\
&= o(\|d_k\|^2),\\
-A_k^{-\top}\nabla f_k &= \lambda_* - A_k^{-\top}(\nabla f_k + A_k^\top \lambda_*)\\
&= \lambda_* + o(1),\\
\nabla f_k^\top e_k &= -(A_k^{-\top}\nabla f_k)^\top c(x_k{+}d_k)\\
&= \lambda_*^\top c(x_k{+}d_k) + o(\|d_k\|^2)\\
&= \frac{1}{2}\lambda_*^\top\left(c''(x_*)\cdot d_k^2\right) + o(\|d_k\|^2),\\
f(x_k{+}d_k{+}e_k) &= f_k + \nabla f_k^\top(d_k + e_k) + \frac{1}{2}d_k^\top \nabla^2 f(x_*) d_k + o(\|d_k\|^2)\\
&= f_k + \nabla f_k^\top d_k + \frac{1}{2}d_k^\top L_* d_k + o(\|d_k\|^2).
\end{aligned}$$

With these estimates, the boundedness of $\{\sigma_k\}$, and the fact that, when there are only equality constraints, the directional derivative of Θ_{σ_k} in the direction d_k can be written $\Theta'_{\sigma_k}(x_k; d_k) = \nabla f_k^\top d_k - \sigma_k\|c_k\|_P$ (see the proof of lemma 17.4), one gets

$$\begin{aligned}
&\Theta_{\sigma_k}(x_k{+}d_k{+}e_k) - \Theta_{\sigma_k}(x_k) - \omega\Theta'_{\sigma_k}(x_k; d_k)\\
&= \nabla f_k^\top d_k + \frac{1}{2}d_k^\top L_* d_k - \sigma_k\|c_k\|_P - \omega\Theta'_{\sigma_k}(x_k; d_k) + o(\|d_k\|^2)\\
&= (1-\omega)\Theta'_{\sigma_k}(x_k; d_k) + \frac{1}{2}d_k^\top L_* d_k + o(\|d_k\|^2). \qquad (17.36)
\end{aligned}$$

We have to show that the right-hand side of (17.36) is nonpositive asymptotically.

Using the optimality conditions of (17.30), the Cauchy-Schwarz inequality (16.14), and the bounds in (17.35), the directional derivative $\Theta'_{\sigma_k}(x_k; d_k) = \nabla f_k^\top d_k - \sigma_k\|c_k\|_P$ can also be written

$$\Theta'_{\sigma_k}(x_k; d_k) = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}})^\top c_k - \sigma_k\|c_k\|_P \le -d_k^\top M_k d_k. \qquad (17.37)$$

Since $d_k^\top L_* d_k \le d_k^\top L_*^r d_k$ for a nonnegative r, (17.36) becomes with (17.37) and (17.34):

$$\begin{aligned}
&\Theta_{\sigma_k}(x_k+d_k+e_k) - \Theta_{\sigma_k}(x_k) - \omega\Theta'_{\sigma_k}(x_k;d_k)\\
&\le \left(\frac{1}{2}-\omega\right)(-d_k^\top M_k d_k) - \frac{1}{2}d_k^\top(M_k - L_*^r)d_k + o(\|d_k\|^2)\\
&\le \left(\frac{1}{2}-\omega\right)(-d_k^\top M_k d_k) + o(\|d_k\|^2).
\end{aligned}$$

For large k, the right-hand side is nonpositive since, by (17.34) and the positive definiteness of L_*^r, $d_k^\top M_k d_k \ge d_k^\top L_*^r d_k + o(\|d_k\|^2) \ge C\|d_k\|^2$, for some positive constant C and large k. □

The result of proposition 17.7 has many variants. It is usually easy to prove them by adapting the arguments used in the proof above (basically by cleverly combining Taylor expansions of an appropriate order). For example, one can avoid using the Hessian of the augmented Lagrangian by replacing condition (17.34) by

$$d_k^\top P_*^\top (M_k - L_*) P_* d_k \ge o(\|d_k\|^2) + o(\|c_k\|),$$

where P_* denotes a projection operator on $N(A_*)$. The proof of this claim has been left as an exercise.

Computing the correction step e_k can be time-consuming for some applications, since this requires a new evaluation of the constraints at $x_k + d_k$. When x_k is far from a solution, this step can also be very large, perturbing uselessly the SQP step d_k. Therefore meticulous implementations of the line-search SQP algorithm usually have a test for deciding whether e_k must be computed and the arc-search detailed above must be substituted for the less expensive line-search. Counter-example 17.6 has shown that the Maratos effect occurs when x_k is on the constraint manifold. On the other hand, truncation of the unit stepsize is unlikely to occur in the neighborhood of a solution when the transversal part of the step prevails. To see this, observe that when c has its values in $\mathbb{R}^n$, the unit stepsize is accepted along Newton's direction to solve $c(x) = 0$ when one uses $x \mapsto \|c(x)\|_{_P}$ as a merit function (see exercise 17.4). These observations suggest that there may be a danger of small stepsize only when the restoration step is small with respect to the tangent step. The next proposition confirms this viewpoint. It shows that the unit stepsize is accepted asymptotically for the iterations satisfying the inequality

$$\|r_k\| \ge C_{\rm ME}\|t_k\|, \tag{17.38}$$

where $C_{\rm ME}$ is a positive constant and $\|\cdot\|$ is an arbitrary norm. To write this inequality, we have decomposed the full step d_k into $d_k = r_k + t_k$, where the restoration step is written $r_k = -A_k^- c_k$, for some right inverse A_k^- of A_k, and the tangent step $t_k \in R(Z_k^-)$ satisfies $\nabla f_k^\top t_k \le 0$.

Proposition 17.8 (admissibility of the unit step-size at restoration prevailing iterations). *Suppose that f and c are of class C^1 in a neighborhood of a stationary point x_* of (P_E). Let $\{x_k\}$ be a sequence converging to x_* and $d_k = r_k + t_k$, where $r_k = -A_k^- c(x_k)$ and $t_k \in R(Z_k^-)$ satisfies $\nabla f(x_k)^\top t_k \leq 0$. Suppose that $\{A_k^-\}$ and $\{\sigma_k\}$ are bounded, that $\sigma_k \geq \|A_k^{-\top} \nabla f(x_k)\|_D + \bar{\sigma}$ for some constant $\bar{\sigma} > 0$, and that $\omega < 1$. Then, for large indices k for which* (17.38) *holds with a positive constant C_{ME}, one has*

$$\Theta_{\sigma_k}(x_k + d_k) \leq \Theta_{\sigma_k}(x_k) + \omega \Theta'_{\sigma_k}(x_k; d_k).$$

Proof. Here, as we shall see, first-order expansions are sufficient. Using the fact that $d_k = O(\|r_k\|)$ for the considered indices, one has

$$\begin{aligned} f(x_k + d_k) &= f_k + \nabla f_k^\top d_k + o(\|r_k\|) \\ c(x_k + d_k) &= c_k + A_k d_k + o(\|r_k\|) \\ &= o(\|r_k\|). \end{aligned}$$

Therefore, using $\Theta'_{\sigma_k}(x_k; d_k) = \nabla f_k^\top d_k - \sigma_k \|c_k\|_P$ (see the proof of lemma 17.4), $\nabla f_k^\top t_k \leq 0$, $\omega < 1$, $\nabla f_k^\top r_k \leq \|A_k^{-\top} \nabla f_k\|_D \|c_k\|_P$, and $r_k = O(\|c_k\|_P)$:

$$\begin{aligned} &\Theta_{\sigma_k}(x_k + d_k) - \Theta_{\sigma_k}(x_k) - \omega \Theta'_{\sigma_k}(x_k; d_k) \\ &= (1-\omega) \nabla f_k^\top d_k - (1-\omega) \sigma_k \|c_k\|_P + o(\|r_k\|) \\ &\leq (1-\omega) \left(\|A_k^{-\top} \nabla f_k\|_D - \sigma_k \right) \|c_k\|_P + o(\|r_k\|) \\ &\leq -(1-\omega) \bar{\sigma} \|c_k\|_P + o(\|c_k\|_P), \end{aligned}$$

which is negative for large k. □

A consequence of this result is that, optimization codes implementing the second order correction often decide to compute e_k and to do an arc-search, only at iterations where (17.38) does not hold. The constant C_{ME} is determined by heuristics.

Modification of the Merit Function: Nondifferentiable Augmented Lagrangian

Another way of getting the asymptotic admissibility of the unit stepsize is to change the merit function. Remember that d_k is obtained by minimizing a quadratic model of the Lagrangian subject to linearized constraints. Hence, taking

$$\ell_{\mu,\sigma}(x) = f(x) + \mu^\top c(x) + \sigma \|c(x)\|_P$$

as a merit function should be convenient, insofar as μ is close enough to λ_* and σ is small enough. The validity of this intuition is confirmed by proposition 17.9 below.

Beforehand, observe that the problem

$$\begin{cases} \min_x \ f(x) + \mu^\top c(x) \\ c(x) = 0, \quad x \in \Omega \end{cases}$$

is clearly equivalent to (P_E). Now, let x_* be a solution to (P_E), with associated multiplier λ_*. Then x_* is still a solution to the problem above, with associated multiplier $\lambda_* - \mu$. Therefore, the results of § 16.4 imply that $\ell_{\mu,\sigma}$ is exact if

$$\sigma > \|\lambda_* - \mu\|_D.$$

On the other hand, one easily computes

$$\ell'_{\mu,\sigma}(x_k; d_k) = -d_k^\top M_k d_k + (\lambda_k^{\mathrm{QP}} - \mu)^\top c_k - \sigma \|c_k\|_P,$$

which is therefore negative if M_k is positive definite and

$$\sigma \geq \|\lambda_k^{\mathrm{QP}} - \mu\|_D.$$

Figure 17.4 shows the level curves of $\ell_{\mu,\sigma}$ for counter-example 17.6, with

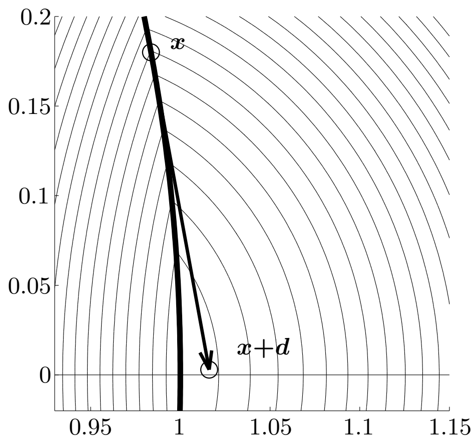

Fig. 17.4. Nondifferentiable augmented Lagrangian

$\tau = 1$, $\mu = -0.55$, and $\sigma = 0.1$.

Proposition 17.9 (admissibility of the unit step-size with a nondifferentiable augmented Lagrangian). *Suppose that f and c are of class C^2 in a neighborhood of a solution x_* to (P_E), satisfying the second-order sufficient conditions of optimality. Let $\{x_k\}$ be a sequence converging to x_*, and d_k be a stationary point of the osculating quadratic problem* (17.30). *In this last problem, suppose that the matrix M_k over-estimates $L_*^r = L_* + rA_*^\top A_*$ in the sense that*

$$d_k^\top (M_k - L_*^r) d_k \geq o(\|d_k\|^2), \tag{17.39}$$

where $r \geq 0$ is such that L_*^r is positive definite (such an r always exists under the assumptions already stated, see exercise 16.1*). Assume also that $d_k \to 0$, that $\omega < \frac{1}{2}$, and that $\sigma_k \geq \|\lambda_k^{\rm QP} - \mu_k\|_{_D}$. Then there exists $\varepsilon > 0$ such that, if $\|\mu_k - \lambda_*\| \leq \varepsilon$ and $0 \leq \sigma_k \leq \varepsilon$, we have for large enough k*

$$\ell_{\mu_k,\sigma_k}(x_k+d_k) \leq \ell_{\mu_k,\sigma_k}(x_k) + \omega \ell'_{\mu_k,\sigma_k}(x_k; d_k).$$

Proof. The following expansions are easily obtained:

$$f(x_k+d_k) = f_k + \nabla f_k^\top d_k + \frac{1}{2} d_k^\top \nabla^2 f(x_*) d_k + o(\|d_k\|^2).$$

$$c(x_k+d_k) = \frac{1}{2} c''(x_*) \cdot d_k^2 + o(\|d_k\|^2).$$

We can then write

$$\begin{aligned}
&\ell_{\mu_k,\sigma_k}(x_k+d_k) - \ell_{\mu_k,\sigma_k}(x_k) - \omega \ell'_{\mu_k,\sigma_k}(x_k; d_k) \\
&= \nabla f_k^\top d_k + \frac{1}{2} d_k^\top \nabla^2 f(x_*) d_k + \frac{1}{2} \mu_k^\top c''(x_*) \cdot d_k^2 - \mu_k^\top c_k - \sigma_k \|c_k\|_{_P} \\
&\quad - \omega \ell'_{\mu_k,\sigma_k}(x_k; d_k) + O(\sigma_k \|d_k\|^2) + o(\|d_k\|^2) \\
&= (1-\omega) \ell'_{\mu_k,\sigma_k}(x_k; d_k) + \frac{1}{2} d_k^\top L_* d_k \\
&\quad + O((\|\mu_k - \lambda_*\|_{_D} + \sigma_k) \|d_k\|^2) + o(\|d_k\|^2) \\
&\leq (1-\omega) \ell'_{\mu_k,\sigma_k}(x_k; d_k) + \frac{1}{2} d_k^\top L_*^r d_k + C_1 \varepsilon \|d_k\|^2 + o(\|d_k\|^2) \\
&\leq \left(\frac{1}{2} - \omega\right) \ell'_{\mu_k,\sigma_k}(x_k; d_k) - \frac{1}{2} d_k^\top (M_k - L_*^r) d_k + C_1 \varepsilon \|d_k\|^2 + o(\|d_k\|^2) \\
&\leq -C_2 \left(\frac{1}{2} - \omega\right) \|d_k\|^2 + C_1 \varepsilon \|d_k\|^2 + o(\|d_k\|^2) \\
&\leq 0,
\end{aligned}$$

if k is large enough and $\varepsilon > 0$ is small enough. We have used the uniform positive definiteness of M_k, which comes from the positive definiteness of L_*^r and from (17.39). □

We refer the reader to the original paper [37] and to [146, 10] for examples of use of the nondifferentiable augmented Lagrangian in implementable algorithms.

17.4 The Hanging Chain Project IV

This is the fourth session dealing with the problem of finding the static equilibrium of chain made of rigid bars that stays above a given tilted floor. The problem was introduced in § 13.8 and developed in §§ 14.7 and 15.4. We now consider the implementation of the globalization technique presented in this chapter. This will provide more robustness to the SQP solver and will give it a tendency to avoid the stationary points that are not local minima.

We propose to use the merit function (17.1) in which $\|\cdot\|_P$ is the ℓ_1 norm $\|v\|_1 := \sum_{i=1}^m |v_i|$:

$$\Theta_\sigma(x) = f(x) + \sigma\|c(x)^\#\|_1. \tag{17.40}$$

This norm satisfies the assumption (17.5) required by proposition 17.1 (see exercise 17.1). The dual norm of the ℓ_1 norm is the ℓ_∞ norm $\|w\|_\infty := \max_{1\leq i\leq m} |w_i|$ (see exercise 16.5).

We assume that the osculating quadratic program has the form (17.2), with a matrix M_k that is symmetric positive definite. This property of M_k is important in order to get a primal solution d_k to (17.2) that is a descent direction of the exact merit function Θ_σ defined by (17.40) (see proposition 17.1). Since the Hessian of the Lagrangian $L_k := \nabla^2_{xx}\ell(x_k, \lambda_k)$ is not necessarily positive definite, we propose to take for M_k a modification of L_k obtained by adding to it a small positive diagonal matrix (using, for example, a modified Cholesky factorization [154, 201]). Using a positive definite quasi-Newton approximation to L_k is another possibility that will be examined in chapter 18.

Modifications to Bring to the `sqp` Function

It is interesting to keep the possibility of using the algorithms defined in the previous sessions by introducing flags. In our code, we use `options.imode (1:2)`, which has the following meanings:

- `imode(1)`: 0 (M_k is a quasi-Newton approximation to L_k), 1 ($M_k = L_k$), 2 ($M_k = L_k + E_k$, where E_k is a small positive diagonal matrix that makes M_k positive definite),
- `imode(2)`: 0 (with line-search), 1 (with unit stepsize).

If we compare the local SQP algorithm on page 257, implemented in the previous sessions, and the version with line-search on page 292, we see that we essentially have to add the steps 3, 4, and 5 of the latter algorithm to the `sqp` function.

- The determination of the penalty parameter σ_k in step 3 can be done by the update rule of page 295. At the first iteration, we take $\sigma_1 = \|\lambda_1^{\mathrm{QP}}\|_D + \bar\sigma$ and set the constant $\bar\sigma$ to $\max(\sqrt{\texttt{eps}}, \|\lambda_1^{\mathrm{QP}}\|_D/100)$.
- The determination of a stepsize α_k along d_k in step 4 can be done like in the backtracking line-search of page 296, with $\beta = 0.1$ and $\alpha_{k,i+1}$ determined by *interpolation*, i.e., as the minimizer of the quadratic function $\alpha \mapsto \xi(\alpha)$ satisfying $\xi(0) = \Theta_{\sigma_k}(x_k)$, $\xi'(0) = \Delta_k$, and $\xi(\alpha_{k,i}) = \Theta_{\sigma_k}(x_k + \alpha_{k,i} d_k)$.

- We set the new multiplier λ_{k+1} by (17.4).

It is better not to limit the number of stepsize trials in the line-search, since this number, which is most often 1, can be large at some difficult iteration. However, the line-search algorithm may cycle when there is an error in the simulator or when rounding errors occur at the end of a minimization. Therefore, some arrangements have to be implemented to prevent this cycling. In our code, the line-search is interrupted when the norm of the step $\alpha_{k,i}\|d_k\|_\infty$ to get improvement in the merit function becomes smaller than a prescribed value `options.dxmin` given on entry in the solver.

It is important to take care over the output printed by the code, since it provides meaningful information on the course of the optimization. Here is the text, in connection with the line-search, that our code prints at each iteration.

```
iter 11,  simul 14,  merit -1.47914e+00,  slope -7.59338e-02
  Armijo's line-search
  1.0000e+00   8.47489e-01   8.47489e-01
  1.0000e-01   1.49986e-03   1.49986e-02
  4.1753e-02  -1.60114e-03  -3.83479e-02
```

The value of Δ_k defined by (17.7), which approximates $\Theta'_{\sigma_k}(x_k;d_k)$, is given after the keyword `slope`, and should always be negative. Each line of the table below the phrase "`Armijo's line-search`" corresponds to a stepsize trial: $\alpha_{k,i}$ is in the first column, $\Theta_{\sigma_k}(x_k+\alpha_{k,i}d_k)-\Theta_{\sigma_k}(x_k)$ in the second, and $(\Theta_{\sigma_k}(x_k+\alpha_{k,i}d_k)-\Theta_{\sigma_k}(x_k))/\alpha_{k,i}$ in the last one. We see in the first column that the unit stepsize $\alpha_{k,1}=1$ is tried first and that it is determined next by interpolation with the safeguard $\beta=0.1$. The last column is useful to detect a possible inconsistency in the simulator (or in the `sqp` function). If d_k is not a descent direction of the merit function Θ_{σ_k} (it should be a descent direction if M_k is positive definite and if nothing is wrong in the simulator and in the `sqp` function, see proposition 17.1), there is a large number of stepsize trials $\alpha_{k,i}$ tending to zero. Then, the value in the last column should tend to Δ_k (this is actually certainly correct if there is no inequality constraint, since then $\Delta_k=\Theta'_{\sigma_k}(x_k;d_k)$, see the comment after proposition 17.1).

Question: Tell why the last value in the third column of the table after the phrase "`Armijo's line-search`" above is often approximately half that of Δ_k (like here: $3.83479/7.59338\simeq 0.505$).

Experimenting with the SQP Algorithm

The first observation is good news: line-search really helps to force convergence. For example, test case 1d (page 249), which diverges without line-search, now converges to the global minimum. Figure 17.5 shows the result with the usual convention: the thin solid bars represent the initial position of the chain, the dashed bars correspond to the intermediate positions, and the

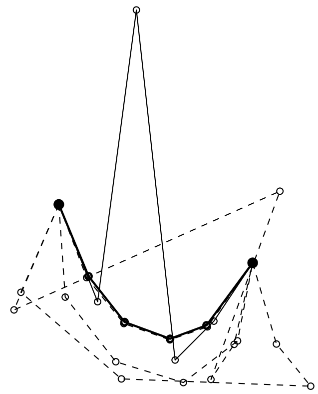

Fig. 17.5. Test case 1d with line-search

bold solid bars are those of the final optimal position. For clarity, we have not represented all the intermediate positions of the 10 iterations required to get convergence, but 1 out of 2.

The second observation is that line-search helps the SQP algorithm to avoid stationary points that are not local minima. For example if we apply the present algorithm with line-search to test case 1b (page 249), the generated sequence now converges to the global minimum of the problem, not to the global maximum as before. The left picture in figure 17.6 shows the result (1

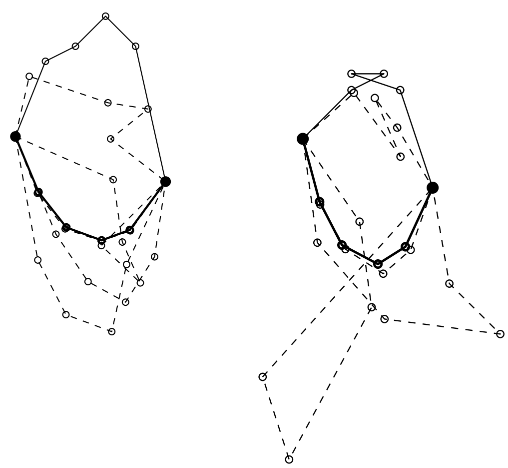

Fig. 17.6. Test cases 1b (left) and 1c (right) with line-search

iteration out of 3). The same phenomenon occurs with test case 1c (page 249), whose convergence to the global minimum is shown in the right hand side picture of figure 17.6.

A third observation: the convergence is smoother with line-search. This is not a very precise concept, but we mean by this that the behavior of the generated sequence is less erratic. Consider for example test case 1f (page 269). The result is shown in figure 17.7. If we compare with figure 15.3, we see that

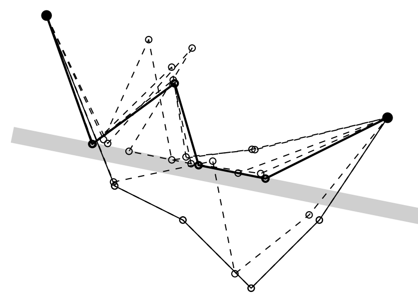

Fig. 17.7. Test case 1f with line-search

the second iterate is now closer the the initial one: the stepsize is actually less than 1 ($\alpha_1 = 0.1$) only at the first iteration. This additional function evaluation is beneficial since the total number of function evaluations is less than the one without line-search (10 instead of 11, not a major improvement, admittedly).

Notes

The use of the exact penalty function (17.1) to globalize the SQP algorithm was proposed by Pshenichnyj (see for example [302]), Han [185; 1977] (with the ℓ_1 norm), and others. The TSQP algorithm described in § 17.2 is taken from [75; 2003]. Another way of dealing with nonconvex problems is to modify the Hessian of the Lagrangian, using a modified Cholesky factorization (see for example [133] and the references therein).

The "effect" described in § 17.3 was discovered by Maratos [247; 1978] and counter-example 17.6 is adapted from [73]. Second-order correction strategies were proposed by Boggs, Tolle, and Wang [36], Coleman and Conn [82], Fletcher [127], Gabay [138], Mayne and Polak [250]. The use of the nondifferentiable augmented Lagrangian was proposed by Bonnans [37]. Note that Fletcher's exact penalty function (16.21) also accepts the unit stepsize asymptotically, but it involves first derivatives, so that its use may lead to expensive algorithms if a number of different stepsizes are required during the line-search or to algorithmic remedies for avoiding expensive operations;

see [299, 33, 34]. Other approaches include the "watchdog" technique [73] and the nonmonotone line-search [281, 46].

To conclude this chapter let us briefly mention and/or review other contributions dealing with the use of second derivatives within SQP, techniques for solving the QP, and algorithmic modifications for tackling large-scale problems: Betts and Frank [29] add a positive multiple of the identity matrix to the full Hessian of the Lagrangian when the factorization of the KKT matrix reveals nonpositive definiteness of the reduced Hessian of the Lagrangian; Bonnans and Launay [45]; Murray and Prieto [270]; Gill, Murray, and Saunders [155]; Leibfritz and Sachs [225]; Facchinei and Lucidi [121]; Boggs, Kearsley, and Tolle [32, 31] propose solving the QP by an interior point method that can be prematurely halted by a pseudo-trust-region constraint, although their method uses line-search for its globalization; Sargent and Ding [321] also use an interior point method to solve the QP inexactly within a line-search approach, but discard the Hessian of the Lagrangian if it fails to yield a descent direction of the merit function; Byrd, Gilbert, and Nocedal [65] combine SQP with an interior point approach on the nonlinear problem and use trust regions for the globalization.

Exercises

17.1. *Norm assumptions.* Let $\|\cdot\|$ be an arbitrary norm on $\mathbb{R}^m$ and consider the following properties (the operators $|\cdot|$ and $(\cdot)^+$ act componentwise; the statements are valid for all u and $v \in \mathbb{R}^m$ when this makes sense):

- (i) $\|\,|u|\,\| = \|u\|$;
- (ii) $|u| \leq |v| \implies \|u\| \leq \|v\|$;
- (iii) $u_i = v_i$ or $0 \implies \|u\| \leq \|v\|$;
- (iv) $0 \leq u \leq v \implies \|u\| \leq \|v\|$;
- (v) $u \leq v \implies \|u^+\| \leq \|v^+\|$;
- (vi) $v \mapsto \|v^+\|$ is convex.

Show that $(i) \Leftrightarrow (ii) \Rightarrow (iii) \Rightarrow (iv) \Leftrightarrow (v) \Leftrightarrow (vi)$, but that none of the other implications holds in general. Show that (vi) may not hold for an arbitrary norm.

Remark: These implications show that assumptions (16.15) and (17.5) on the norm $\|\cdot\|_P$ are satisfied with the ℓ_p norms, $1 \leq p \leq \infty$, since ℓ_p norms satisfy (i). They also show that (16.15) is more restrictive than (17.5).

17.2. *On the directional derivative of Θ_σ.* Find a one-dimensional example, in which $\Theta'_\sigma(x;d) < \nabla f(x)^\top d - \sigma\|c(x)^\#\|_P$, where d is the solution to the osculating quadratic problem (17.2) (hence the inequality in (17.6) may be strict).

[*Hint*: Equality holds if $I = \emptyset$.]

17.3. *Descent direction for the exact penalization of the Lagrangian.* Consider the exact penalty function $\Theta_{\mu,\sigma} : \mathbb{R}^n \to \mathbb{R}$ defined for $\mu \in \mathbb{R}^m$ and $\sigma > 0$ by

$$\Theta_{\mu,\sigma}(x) := f(x) + \mu^\top c(x)^\# + \sigma \|c(x)^\#\|_P,$$

where the norm $\|\cdot\|_P$ satisfies (17.5) (see also exercise 16.9). Let $(d_k, \lambda_k^{\mathrm{QP}})$ satisfy the optimality conditions (17.3). Show that d_k is a descent direction of $\Theta_{\mu,\sigma}$ at x_k, provided x_k is not a stationary point of (P_{EI}), M_k is positive definite, $\sigma \geq \|\lambda_k^{\mathrm{QP}} - \mu\|_D$, and $\mu_I \geq 0$.

17.4. *Admissibility of the unit stepsize for Newton's method.* Consider the problem of finding a root x_* of the equation $F(x) = 0$, where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. Newton's method consists in updating x by $x_+ = x + d$, where d solves $F'(x)d = -F(x)$ (see § 13.7). Let $\|\cdot\|$ be an arbitrary norm and consider $\varphi(x) = \|F(x)\|$ as a merit function for this problem. Suppose that $F'(x_*)$ is nonsingular. Show that, for any constant $\omega \in]0,1[$, there is a neighborhood V of x_*, such that if $x \in V$, $\varphi(x+d) \leq \varphi(x) + \omega\varphi'(x;d)$.

References

1. I. Adler, M.G.C. Resende, and G. Veiga. An implementation of Karmarkar's algorithm for linear programming. *Math. Programming*, 44:297–335, 1989.
2. M. Al-Baali. Descent property and global convergence of the Fletcher-Reeves methods with inexact line search. *IMA Journal of Numerical Analysis*, 5:121–124, 1985.
3. W. Alt. The Lagrange-Newton method for infinite-dimensional optimization problems. *Numerical Functional Analysis and Optimization*, 11:201–224, 1990.
4. W. Alt. Sequential quadratic programming in Banach spaces. In W. Oettli and D. Pallaschke, editors, *Advances in Optimization*, number 382 in Lecture Notes in Economics and Mathematical Systems, pages 281–301. Springer-Verlag, 1992.
5. W. Alt. Semi-local convergence of the Lagrange-Newton method with application to optimal control. In R. Durier and Ch. Michelot, editors, *Recent Developments in Optimization*, number 429 in Lecture Notes in Economics and Mathematical Systems, pages 1–16. Springer-Verlag, 1995.
6. W. Alt and K. Malanowski. The Lagrange-Newton method for nonlinear optimal control problems. *Computational Optimization and Applications*, 2:77–100, 1991.
7. M. Anitescu. Degenerate nonlinear programming with a quadratic growth condition. *SIAM Journal on Optimization*, 10:1116–1135, 2000.
8. M. Anitescu. On the rate of convergence of sequential quadratic programming with nondifferentiable exact penalty function in the presence of constraint degeneracy. *Mathematical Programming*, 92:359–386, 2002.
9. K.M. Anstreicher and M.H. Wright. A note on the augmented Hessian when the reduced Hessian is semidefinite. *SIAM Journal on Optimization*, 11:243–253, 2000.
10. P. Armand and J.Ch. Gilbert. A piecewise line-search technique for maintaining the positive definiteness of the updated matrices in the SQP method. *Computational Optimization and Applications*, 16:121–158, 2000.
11. P. Armand, J.Ch. Gilbert, and S. Jan-Jégou. A BFGS-IP algorithm for solving strongly convex optimization problems with feasibility enforced by an exact penalty approach. *Mathematical Programming*, 92:393–424, 2002.
12. L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966.
13. K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Nonlinear Programming.* Stanford University Press, Stanford, CA, 1958.
14. K.J. Arrow and R.M. Solow. Gradient methods for constrained maxima with weakened assumptions. In K.J. Arrow, L. Hurwicz, and H. Uzawa, editors,

Studies in Linear and Nonlinear Programming. Stanford University Press, Standford, Calif., 1958.

15. A. Auslender. Numerical methods for nondifferentiable convex optimization. *Mathematical Programming Study*, 30:102–126, 1987.
16. A. Auslender and M. Teboulle. Lagrangian duality and related multiplier methods for variational inequality problems. *SIAM Journal on Optimization*, 10:1097–1115, 2000.
17. A. Auslender, M. Teboulle, and S. Ben-Tiba. Interior proximal and multiplier methods based on second order homogeneous functionals. *Mathematics of Operations Research*, 24:645–668, 1999.
18. F. Babonneau, C. Beltran, A. Haurie, C. Tadonki, and J.-P. Vial. Proximal-ACCPM: a versatile oracle based optimization method. To appear in *Computational Management Science*, 2006.
19. L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal. Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners. *Computational Optimization and Applications*, 20(3):227–244, 2001.
20. R.E. Bank, B.D. Welfert, and H. Yserentant. A class of iterative methods for solving saddle point problems. *Numerische Mathematik*, 56:645–666, 1990.
21. A. Belloni, A. Diniz, M.E. Maceira, and C. Sagastizábal. Bundle relaxation and primal recovery in Unit Commitment problems. The Brazilian case. *Annals of Operations Research*, 120:21–44, 2003.
22. A. Belloni and C. Sagastizábal. Dynamic bundle methods: Application to combinatorial optimization. Technical report, Optimization on line, 2004. `http://www.optimization-online.org/DB_HTML/2004/08/925.html`.
23. A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization – Analysis, Algorithms, and Engineering Applications*. MPS/SIAM Series on Optimization 2. SIAM, 2001.
24. M.W. Berry, M.T. Health, I. Kaneko, M. Lawo, R.J. Plemmons, and R.C. Ward. An algorithm to compute a sparse basis of the null space. *Numerische Mathematik*, 47:483–504, 1985.
25. D.P. Bertsekas. Multiplier methods: a survey. *Automatica*, 12:133–145, 1976.
26. D.P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.
27. D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995. Second edition, 1999.
28. J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, 2001.
29. J.T. Betts and P.D. Frank. A sparse nonlinear optimization algorithm. *Journal of Optimization Theory and Applications*, 3:519–541, 1994.
30. L.T. Biegler, J. Nocedal, and C. Schmid. A reduced Hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, 5:314–347, 1995.
31. P.T. Boggs, A.J. Kearsley, and J.W. Tolle. A global convergence analysis of an algorithm for large-scale nonlinear optimization problems. *SIAM Journal on Optimization*, 9:833–862, 1999.
32. P.T. Boggs, A.J. Kearsley, and J.W. Tolle. A practical algorithm for general large scale nonlinear optimization problems. *SIAM Journal on Optimization*, 9:755–778, 1999.

33. P.T. Boggs and J.W. Tolle. A family of descent functions for constrained optimization. *SIAM Journal on Numerical Analysis*, 21:1146–1161, 1984.
34. P.T. Boggs and J.W. Tolle. A strategy for global convergence in a sequential quadratic programming algorithm. *SIAM Journal on Numerical Analysis*, 26:600–623, 1989.
35. P.T. Boggs and J.W. Tolle. Sequential quadratic programming. In *Acta Numerica 1995*, pages 1–51. Cambridge University Press, 1995.
36. P.T. Boggs, J.W. Tolle, and P. Wang. On the local convergence of quasi-Newton methods for constrained optimization. *SIAM Journal on Control and Optimization*, 20:161–171, 1982.
37. J.F. Bonnans. Asymptotic admissibility of the unit stepsize in exact penalty methods. *SIAM Journal on Control and Optimization*, 27:631–641, 1989.
38. J.F. Bonnans. Local study of Newton type algorithms for constrained problems. In S. Dolecki, editor, *Optimization*, number 1405 in Lecture Notes in Mathematics, pages 13–24. Springer-Verlag, 1989.
39. J.F. Bonnans. Théorie de la pénalisation exacte. *Modélisation Mathématique et Analyse Numérique*, 24:197–210, 1990.
40. J.F. Bonnans. Local analysis of Newton-type methods for variational inequalities and nonlinear programming. *Applied Mathematics and Optimization*, 29:161–186, 1994.
41. J.F. Bonnans. Exact penalization with a small nonsmooth term. *Revista de Matemáticas Aplicadas*, 17:37–45, 1996.
42. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. A family of variable metric proximal methods. *Mathematical Programming*, 68:15–47, 1995.
43. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Optimisation Numérique – Aspects théoriques et pratiques*. Number 27 in Mathématiques et Applications. Springer Verlag, Berlin, 1997.
44. J.F. Bonnans and C.C. Gonzaga. Convergence of interior point algorithms for the monotone linear complementarity problem. *Mathematics of Operations Research*, 21:1–25, 1996.
45. J.F. Bonnans and G. Launay. Sequential quadratic programming with penalization of the displacement. *SIAM Journal on Optimization*, 5:792–812, 1995.
46. J.F. Bonnans, E.R. Panier, A.L. Tits, and J.L. Zhou. Avoiding the Maratos effect by means of a nonmonotone line search II: Inequality constrained problems – Feasible iterates. *SIAM Journal on Numerical Analysis*, 29:1187–1202, 1992.
47. J.F. Bonnans, C. Pola, and R. Rebaï. Perturbed path following interior point algorithms. *Optimization Methods and Software*, 11-12:183–210, 1999.
48. J.F. Bonnans and F.A. Potra. Infeasible path following algorithms for linear complementarity problems. *Mathematics of Operations Research*, 22:378–407, 1997.
49. J.F. Bonnans and A. Shapiro. Optimization problems with perturbations – A guided tour. *SIAM Review*, 40:202–227, 1998.
50. J.F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Verlag, New York, 2000.
51. W. Boothby. *An Introduction to Differentiable Manifolds and Differential Geometry*. Academic Press, New York, 1975.

52. J. Borwein and A.S. Lewis. *Convex Analysis and Nonlinear Optimization*. Springer Verlag, New York, 2000.
53. J.H. Bramble and J.E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation*, 50:1–7, 1988.
54. J.H. Bramble, J.E. Pasciak, and A.T. Vassilev. Analysis of the Uzawa algorithm for saddle point problems. *SIAM Journal on Numerical Analysis*, 34:1072–1092, 1997.
55. U. Brännlund. *On relaxation methods for nonsmooth convex optimization*. PhD thesis, Royal Institute of Technology - Stockholm, 1993.
56. J.R. Bunch and L. Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Mathematics of Computation*, 31:163–179, 1977.
57. R.S. Burachik, A.N. Iusem, and B.F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Anal.*, 5(2):159–180, 1997.
58. R.S. Burachik, C. Sagastizábal, and S. Scheinberg de Makler. An inexact method of partial inverses and a parallel bundle method. *Optimization Methods and Software*, 21(3):385–400, 2006.
59. R.S. Burachik, C. Sagastizábal, and B. F. Svaiter. Bundle methods for maximal monotone operators. In R. Tichatschke and M. Théra, editors, *Ill-posed Variational Problems and Regularization Techniques*, number 477 in Lecture Notes in Economics and Mathematical Systems, pages 49–64. Springer-Verlag Berlin Heidelberg, 1999.
60. J.V. Burke. An exact penalization viewpoint of constrained optimization. *SIAM Journal on Control and Optimization*, 29:968–998, 1991.
61. J.V. Burke and S.-P. Han. A robust sequential quadratic programming method. *Mathematical Programming*, 43:277–303, 1989.
62. J.D. Buys. *Dual algorithms for constrained optimization*. PhD thesis, Rijksuniversiteit te Leiden, Leiden, The Netherlands, 1972.
63. R.H. Byrd. An example of irregular convergence in some constrained optimization methods that use the projected Hessian. *Mathematical Programming*, 32:232–237, 1985.
64. R.H. Byrd. On the convergence of constrained optimization methods with accurate Hessian information on a subspace. *SIAM Journal on Numerical Analysis*, 27:141–153, 1990.
65. R.H. Byrd, J.Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89:149–185, 2000.
66. R.H. Byrd and J. Nocedal. A tool for the analysis of quasi-Newton methods with application to unconstrained minimization. *SIAM Journal on Numerical Analysis*, 26:727–739, 1989.
67. R.H. Byrd and J. Nocedal. An analysis of reduced Hessian methods for constrained optimization. *Mathematical Programming*, 49:285–323, 1991.
68. R.H. Byrd and R.B. Schnabel. Continuity of the null space basis and constrained optimization. *Mathematical Programming*, 35:32–41, 1986.
69. R.H. Byrd, R.A. Tapia, and Y. Zhang. An SQP augmented Lagrangian BFGS algorithm for constrained optimization. *SIAM Journal on Optimization*, 2:210–241, 1992.

70. A. Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *C. R. Acad. Sci. Paris*, 25:535–538, 1847.
71. J. Céa. *Optimisation: Théorie et Algorithmes*. Dunod, Paris, 1971.
72. Y. Chabrillac and J.-P. Crouzeix. Definiteness and semidefiniteness of quadratic forms revisited. *Linear Algebra and its Applications*, 63:283–292, 1984.
73. R.M. Chamberlain, C. Lemaréchal, H.C. Pedersen, and M.J.D. Powell. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Study*, 16:1–17, 1982.
74. C. Charalambous. A lower bound for the controlling parameters of the exact penalty functions. *Mathematical Programming*, 15:278–290, 1978.
75. L. Chauvier, A. Fuduli, and J.Ch. Gilbert. A truncated SQP algorithm for solving nonconvex equality constrained optimization problems. In G. Di Pillo and A. Murli, editors, *High Performance Algorithms and Software for Nonlinear Optimization*, pages 146–173. Kluwer Academic Publishers B.V., 2003.
76. G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Math. Programming*, 64(1, Ser. A):81–101, 1994.
77. X. Chen and M. Fukushima. Proximal quasi-Newton methods for nondifferentiable convex optimization. *Math. Program.*, 85(2, Ser. A):313–334, 1999.
78. E. Cheney and A. Goldstein. Newton's method for convex programming and Tchebycheff approximations. *Numerische Mathematik*, 1:253–268, 1959.
79. P.G. Ciarlet. *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation* (second edition). Masson, Paris, 1988.
80. F.H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York; reprinted by SIAM, 1983.
81. T.F. Coleman and A.R. Conn. Nonlinear programming via an exact penalty function: asymptotic analysis. *Mathematical Programming*, 24:123–136, 1982.
82. T.F. Coleman and A.R. Conn. Nonlinear programming via an exact penalty function: global analysis. *Mathematical Programming*, 24:137–161, 1982.
83. T.F. Coleman and D.C. Sorensen. A note on the computation of an orthonormal basis for the null space of a matrix. *Mathematical Programming*, 29:234–242, 1984.
84. L. Conlon. *Differentiable Manifolds – A first Course*. Birkhauser, Boston, 1993.
85. A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*. Number 17 in Computational Mathematics. Springer Verlag, Berlin, 1992.
86. A.R. Conn, N.I.M. Gould, and Ph.L. Toint. *Trust-Region Methods*. MPS/SIAM Series on Optimization. MPS/SIAM, Philadelphia, 2000.
87. D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplications. *SIAM J. Computation*, 11:472–492, 1982.
88. G. Corliss and A. Griewank, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. Proceedings in Applied Mathematics 53. SIAM, Philadelphia, 1991.
89. R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62:261–275, 1993.
90. R.W. Cottle. Manifestations of the Schur complement. *Linear Algebra and its Applications*, 8:189–211, 1974.
91. R.W. Cottle, J.S. Pang, and R.E. Stone. *The linear complementarity problem*. Academic Press, New York, 1992.

92. M. Cui. A sufficient condition for the convergence of the inexact Uzawa algorithm for saddle point problems. *Journal of Computational and Applied Mathematics*, 139:189–196, 2002.
93. J.-C. Culioli and G. Cohen. Decomposition/coordination algorithms in stochastic optimization. *SIAM J. Control Optim.*, 28(6):1372–1403, 1990.
94. G.B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, N.J., 1963.
95. G.B. Dantzig and P. Wolfe. The decomposition algorithm for linear programming. *Econometrica*, 29(4):767–778, 1961.
96. W.C. Davidon. Variable metric methods for minimization. AEC Research and Development Report ANL-5990, Argonne National Laboratory, Argonne, Illinois, 1959.
97. W.C. Davidon. Variable metric method for optimization. *SIAM Journal on Optimization*, 1:1–17, 1991.
98. F. Delbos and J.Ch. Gilbert. Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems. *Journal of Convex Analysis*, 12:45–69, 2005.
99. F. Delbos, J.Ch. Gilbert, R. Glowinski, and D. Sinoquet. Constrained optimization in seismic reflection tomography: a Gauss-Newton augmented Lagrangian approach. *Geophysical Journal International*, 164:670–684, 2006.
100. F. Delprat-Jannaud and P. Lailly. What information on the Earth model do reflection travel times hold ? *Journal of Geophysical Research*, 97:19827–19844, 1992.
101. F. Delprat-Jannaud and P. Lailly. Ill-posed and well-posed formulations of the reflection travel time tomography problem. *Journal of Geophysical Research*, 98:6589–6605, 1993.
102. R.S. Dembo and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26:190–212, 1983.
103. D. den Hertog. *Interior-point approach to linear, quadratic and convex programming*. Kluwer Academic Publishers, Boston, 1994.
104. J.E. Dennis and J.J. Moré. A characterization of superlinear convergence and its application to quasi-Newton methods. *Mathematics of Computation*, 28:549–560, 1974.
105. J.E. Dennis and J.J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.
106. J.E. Dennis and R.B. Schnabel. A new derivation of symmetric positive definite secant updates. In *Nonlinear Programming 4*, pages 167–199. Academic Press, 1981.
107. J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, 1983.
108. P. Deuflhard. *Newton Methods for Nonlinear Problems – Affine Invariance and Adaptative Algorithms*. Number 35 in Computational Mathematics. Springer, Berlin, 2004.
109. G. Di Pillo and L. Grippo. A new class of augmented Lagrangians in nonlinear programming. *SIAM Journal on Control and Optimization*, 17:618–628, 1979.
110. G. Di Pillo and S. Lucidi. On exact augmented Lagrangian functions in nonlinear programming. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, pages 85–100. Plenum Press, New York, 1996.
111. G. Di Pillo and S. Lucidi. An augmented Lagrangian function with improved exactness properties. *SIAM Journal on Optimization*, 12:376–406, 2001.

112. M.P. do Carmo. *Riemannian Geometry*. Birkhauser, Boston, 1993.
113. E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91:201–213, 2002.
114. I.S. Duff and J.K. Reid. MA27 – A set of Fortran subroutines for solving sparse symmetric sets of linear equations. Technical Report AERE R10533, HMSO, London, 1982.
115. I.S. Duff and J.K. Reid. The multifrontal solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, 9:301–325, 1983.
116. I.S. Duff and J.K. Reid. Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, 22:227–257, 1996.
117. I. Ekeland and R. Temam. *Analyse convexe et problèmes variationnels*. Dunod-Gauthier Villars, Paris, 1974.
118. H.C. Elman and G.H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM Journal on Numerical Analysis*, 31:1645–1661, 1994.
119. I.I. Eremin. The penalty method in convex programming. *Soviet Mathematics Doklady*, 8:459–462, 1966.
120. F. Facchinei. Exact penalty functions and Lagrange multipliers. *Optimization*, 22:579–606, 1991.
121. F. Facchinei and S. Lucidi. Convergence to second-order stationary points in inequality constrained optimization. *Mathematics of Operations Research*, 23:746–766, 1998.
122. A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: sequential unconstrained minimization technique*. J. Wiley, New York, 1968.
123. P. Finsler. Über das vorkommen definiter und semidefiniter formen und scharen quadratischer formen. *Commentarii Mathematici Helvetica*, 9:188–192, 1937.
124. R. Fletcher. A class of methods for nonlinear programming with termination and convergence properties. In J. Abadie, editor, *Integer and Nonlinear Programming*. North-Holland, Amsterdam, 1970.
125. R. Fletcher. A FORTRAN subroutine for quadratic programming. Report R 6370, Atomic Energy Research Establishment, Harwell, England, 1970.
126. R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Study*, 17:67–76, 1982.
127. R. Fletcher. Second order corrections for non-differentiable optimization. In D. Griffiths, editor, *Numerical Analysis*, pages 85–114. Springer-Verlag, 1982.
128. R. Fletcher. *Practical Methods of Optimization* (second edition). John Wiley & Sons, Chichester, 1987.
129. R. Fletcher and T. Johnson. On the stability of null-space methods for KKT systems. *SIAM Journal on Matrix Analysis and Applications*, 18:938–958, 1997.
130. R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91:239–269, 2002.
131. R. Fletcher and S. Leyffer. A bundle filter method for nonsmooth nonlinear optimization. Technical report, University of Dundee Numerical Analysis Report NA 195, December, 1999. `http://www-unix.mcs.anl.gov/~leyffer/papers/nsfilter.pdf`.

132. R. Fletcher and M.J.D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6:163–168, 1963.
133. A. Forsgren, P.E. Gill, and W. Murray. Computing modified Newton directions using a partial Cholesky factorization. *SIAM Journal on Scientific Computing*, 16:139–150, 1995.
134. M. Fortin and R. Glowinski. *Méthodes de Lagrangien Augmenté – Applications à la Résolution Numérique de Problèmes aux Limites*. Number 9 in Méthodes Mathématiques de l'Informatique. Dunod, Paris, 1982.
135. A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2003.
136. M. Fukushima. A descent algorithm for nonsmooth convex optimization. *Mathematical Programming*, 30:163–175, 1984.
137. D. Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37:177–219, 1982.
138. D. Gabay. Reduced quasi-Newton methods with feasibility improvement for nonlinearly constrained optimization. *Mathematical Programming Study*, 16:18–44, 1982.
139. Gauss. *Theoria motus corporum coelestium*. F. Perthes and I.H. Besser, Hamburg, 1809.
140. J. Gauvin. A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming. *Mathematical Programming*, 12:136–138, 1977.
141. J. Gauvin. *Théorie de la programmation mathématique non convexe*. Les Publications CRM, Montréal, 1992.
142. J. Gauvin. *Lecons de Programmation Mathématique*. Éditions de l'École Polytechnique de Montréal, Montréal, 1995.
143. D.M. Gay, M.L. Overton, and M.H. Wright. A primal-dual interior method for nonconvex nonlinear programming. In Y.-X. Yuan, editor, *Advances in Nonlinear Programming*. Kluwer Academic Publishers, 1998.
144. J.Ch. Gilbert. Mise à jour de la métrique dans les méthodes de quasi-Newton réduites en optimisation avec contraintes d'égalité. *Modélisation Mathématique et Analyse Numérique*, 22:251–288, 1988.
145. J.Ch. Gilbert. On the local and global convergence of a reduced quasi-Newton method. *Optimization*, 20:421–450, 1989.
146. J.Ch. Gilbert. Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization. *Mathematical Programming*, 50:1–28, 1991.
147. J.Ch. Gilbert. Superlinear convergence of a reduced BFGS method with piecewise line-search and update criterion. Rapport de Recherche 2140, INRIA, BP 105, 78153 Le Chesnay, France, 1993.
148. J.Ch. Gilbert. On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization. *SIAM Journal on Optimization*, 7:780–813, 1997.
149. J.Ch. Gilbert. Piecewise line-search techniques for constrained minimization by quasi-Newton algorithms. In Y.-X. Yuan, editor, *Advances in Nonlinear Programming*, chapter 4, pages 73–103. Kluwer Academic Publishers, 1998.
150. J.Ch. Gilbert. *Éléments d'Optimisation Différentiable – Théorie et Algorithmes*. 2006. `http://www-rocq.inria.fr/~gilbert/ensta/optim.html`.

151. J.Ch. Gilbert, G. Le Vey, and J. Masse. La différentiation automatique de fonctions représentées par des programmes. Rapport de Recherche n° 1557, Inria, BP 105, F-78153 Le Chesnay, France, 1991.
152. J.Ch. Gilbert and C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Mathematical Programming*, 45:407–435, 1989.
153. J.Ch. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2:21–42, 1992.
154. P.E. Gill and W. Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, 7:311–350, 1974.
155. P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: an SQP algorithm for large-scale constrained optimization. Numerical Analysis Report 96-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1996.
156. P.E. Gill, W. Murray, and M.A. Saunders. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12:979–1006, 2002.
157. P.E. Gill, W. Murray, M.A. Saunders, G.W. Stewart, and M.H. Wright. Properties of a representation of a basis for the null space. *Mathematical Programming*, 33:172–186, 1985.
158. P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. User's guide for NPSOL (version 4.0): a Fortran package for nonlinear programming. Technical Report SOL-86-2, Department of Operations Research, Stanford University, Stanford, CA 94305, 1986.
159. P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. Constrained nonlinear programming. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Handbooks in Operations Research and Management Science*, volume 1: Optimization, chapter 3, pages 171–210. Elsevier Science Publishers B.V., North-Holland, 1989.
160. P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, New York, 1981.
161. S.T. Glad. Properties of updating methods for the multipliers in augmented Lagrangians. *Journal of Optimization Theory and Applications*, 28:135–156, 1979.
162. R. Glowinski and Q.-H. Tran. Constrained optimization in reflexion tomography: the augmented Lagrangian method. *East-West J. Numer. Math.*, 1(3):213–234, 1993.
163. J.-L. Goffin. On convergence rates of subgradient optimization methods. *Mathematical Programming*, 13:329–347, 1977.
164. J.L. Goffin and K.C. Kiwiel. Convergence of a simple sugradient level method. *Math. Program.*, 85:207–211, 1999.
165. D. Goldfarb and A. Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.
166. D. Goldfarb and M.J. Todd. Linear programming. In G.L. Nemhauser et al., editor, *Handbook on Operations Research and Management Science*, volume 1, Optimization, pages 73–170. North-Holland, 1989.

167. A.J. Goldman and A.W. Tucker. Polyhedral convex cones. In H.W. Kuhn and A.W. Tucker, editors, *Linear inequalities and related systems*, pages 19–40, Princeton, 1956. Princeton University Press.
168. A.A. Goldstein and J.F. Price. An effective algorithm for minimization. *Numerische Mathematik*, 10:184–189, 1967.
169. E.G. Gol'shteĭn and N.V. Tretyakov. *Modified Lagrangians and Monotone Maps in Optimization*. Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1996.
170. G.H. Golub and C.F. Van Loan. *Matrix Computations* (second edition). The Johns Hopkins University Press, Baltimore, Maryland, 1989.
171. C.C. Gonzaga. Polynomial affine algorithms for linear programming. *Math. Programming*, 49:7–21, 1990.
172. C.C. Gonzaga. Path following methods for linear programming. *SIAM Review*, 34:167–227, 1992.
173. C.C. Gonzaga. A simple presentation of Karmarkar's algorithm. In *Workshop on interior point methods*, Budapest, 1993.
174. C.C. Gonzaga. The largest step path following algorithm for monotone linear complementarity problems. *Mathematical Programming*, 76:309–332, 1997.
175. C.C. Gonzaga and J.F. Bonnans. Fast convergence of the simplified largest step path following algorithm. *Mathematical Programming series B*, 76:95–115, 1997.
176. C.C. Gonzaga and R.A. Tapia. On the convergence of the Mizuno–Todd–Ye algorithm to the analytic center of the solution set. *SIAM J. Optimization*, 7:47–65, 1997.
177. J. Goodman. Newton's method for constrained optimization. *Mathematical Programming*, 33:162–171, 1985.
178. N. Gould, D. Orban, and Ph.L. Toint. Numerical methods for large-scale nonlinear optimization. In *Acta Numerica 2005*, pages 299–361. Cambridge University Press, 2005.
179. N.I.M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32:90–99, 1985.
180. G. Gramlich, R. Hettich, and E.W. Sachs. Local convergence of SQP methods in semi-infinite programming. *SIAM Journal on Optimization*, 5:641–658, 1995.
181. A. Griewank. *Evaluating Derivatives – Principles and Techniques of Algorithmic Differentiation*. SIAM Publication, 2000.
182. C.B. Gurwitz. Local convergence of a two-piece update of a projected Hessian matrix. *SIAM Journal on Optimization*, 4:461–485, 1994.
183. W.W. Hager. Stabilized sequential quadratic programming. *Computational Optimization and Applications*, 12:253–273, 1999.
184. S.-P. Han. Superlinearly convergent variable metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.
185. S.-P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22:297–309, 1977.
186. S.-P. Han and O.L. Mangasarian. Exact penalty functions in nonlinear programming. *Mathematical Programming*, 17:251–269, 1979.
187. P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.

188. M. Held and R. Karp. The traveling salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1(1):6–25, 1971.
189. J. Herskovits. A view on nonlinear optimization. In J. Herskovits, editor, *Advances in Structural Optimization*, pages 71–116. Kluwer Academic Publishers, 1995.
190. J. Herskovits. Feasible direction interior-point technique for nonlinear optimization. *Journal of Optimization Theory and Applications*, 99:121–146, 1998.
191. M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969.
192. M.R. Hestenes. *Conjugate Direction Methods in Optimization.* Number 12 in Applications of Mathematics. Springer-Verlag, 1980.
193. M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
194. N.J. Higham. *Accuracy and Stability of Numerical Algorithms* (second edition). SIAM Publication, Philadelphia, 2002.
195. J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms.* Number 305-306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, Berlin, 1993.
196. J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis.* Springer-Verlag, Berlin, 2001. Abridged version of *Convex analysis and minimization algorithms. I* and *II* [Springer, Berlin, 1993].
197. W. Hoyer. Variants of the reduced Newton method for nonlinear equality constrained optimization problems. *Optimization*, 17:757–774, 1986.
198. A.D. Ioffe. Necessary and sufficient conditions for a local minimum. 1: a reduction theorem and first order conditions. *SIAM Journal on Control and Optimization*, 17:245–250, 1979.
199. B. Jansen, C. Roos, and T. Terlaky. The theory of linear programming: skew symmetric self-dual problems and the central path. *Optimization*, 29:225–233, 1994.
200. A.H.G. Rinnooy Kan and G.T. Timmer. Global optimization. In G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, editors, *Handbooks in Operations Research and Management Science*, volume 1: Optimization, chapter 9, pages 631–662. Elsevier Science Publishers B.V., North-Holland, 1989.
201. S. Kaniel and A. Dax. A modified Newton's method for unconstrained minimization. *SIAM Journal on Numerical Analysis*, 16:324–331, 1979.
202. L.V. Kantorovich and G.P. Akilov. *Functional Analysis* (second edition). Pergamon Press, London, 1982.
203. E. Karas, A. Ribeiro, C. Sagastizábal, and M.V. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Math. Program. Ser. B*, 2006. Accepted for publication.
204. N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.
205. J. E. Kelley. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.
206. L. Khachian. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.
207. L.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Adad. Nauk SSSR*, 244:1093–1096, 1979. Trad. anglaise : Soviet Math. Doklady 20(1979), 191-194.

208. K.V. Kim, Yu.E. Nesterov, and B.V. Cherkasskiĭ. An estimate of the effort in computing the gradient. *Soviet Math. Dokl.*, 29:384–387, 1984.
209. K.C. Kiwiel. An exact penalty function algorithm for nonsmooth convex constrained minimization problems. *IMA J. Numer. Anal.*, 5(1):111–119, 1985.
210. K.C. Kiwiel. *Methods of Descent for Nondifferentiable Optimization.* Lecture Notes in Mathematics 1133. Springer Verlag, Berlin, 1985.
211. K.C. Kiwiel. A constraint linearization method for nondifferentiable convex minimization. *Numerische Mathematik*, 51:395–414, 1987.
212. K.C. Kiwiel. A subgradient selection method for minimizing convex functions subject to linear constraints. *Computing*, 39(4):293–305, 1987.
213. K.C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46(1):105–122, 1990.
214. K.C. Kiwiel. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Math. Programming*, 52(2, Ser. B):285–302, 1991.
215. K.C. Kiwiel, T. Larsson, and P.O. Lindberg. The efficiency of ballstep subgradient level methods for convex optimization. *Mathematics of Operations Research*, 24(1):237–254, 1999.
216. V. Klee and G.L. Minty. How good is the simplex algorithm ? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, New York, 1972.
217. M. Kojima, N. Megiddo, T. Noma, and A. Yoshise. *A unified approach to interior point algorithms for linear complementarity problems.* Number 538 in Lecture Notes in Computer Science. Springer Verlag, Berlin, 1991.
218. K. Kortanek and Z. Jishan. New purification algorithms for linear programming. *Naval Research Logistics*, 35:571–583, 1988.
219. F.-S. Kupfer. An infinite-dimensional convergence theory for reduced SQP methods in Hilbert space. *SIAM Journal on Optimization*, 6:126–163, 1996.
220. J. Kyparisis. On uniqueness of Kuhn-Tucker multipliers in nonlinear programming. *Mathematical Programming*, 32:242–246, 1985.
221. L. Lasdon. *Optimization Theory for Large Systems.* Macmillan Series in Operations Research, 1970.
222. C.T. Lawrence and A.L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6:265–282, 1996.
223. C.T. Lawrence and A.L. Tits. Feasible sequential quadratic programming for finely discretized problems from SIP. In R. Reemtsen and J.-J. Rückmann, editors, *Semi-infinite Programming*, pages 159–193. Kluwer Academic Publishers B.V., 1998.
224. C.T. Lawrence and A.L. Tits. A computationally efficient feasible sequential quadratic programming algorithm. *SIAM Journal on Optimization*, 11:1092–1118, 2001.
225. F. Leibfritz and E.W. Sachs. Inexact SQP interior point methods and large scale optimal control problems. *SIAM Journal on Optimization*, 38:272–293, 1999.
226. C. Lemaréchal. An algorithm for minimizing convex functions. In J.L. Rosenfeld, editor, *Information Processing '74*, pages 552–556. North Holland, 1974.
227. C. Lemaréchal. An extension of Davidon methods to nondifferentiable problems. *Mathematical Programming Study*, 3:95–109, 1975.

228. C. Lemaréchal. A view of line-searches. In A. Auslender, W. Oettli, and J. Stoer, editors, *Optimization and Optimal Control*, number 30 in Lecture Notes in Control and Information Science, pages 59–78. Springer-Verlag, Heidelberg, 1981.
229. C. Lemaréchal and R. Mifflin. Global and superlinear convergence of an algorithm for one-dimensional minimzation of convex functions. *Mathematical Programming*, 24:241–256, 1982.
230. C. Lemaréchal, A.S. Nemirovskii, and Yu.E. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69:111–148, 1995.
231. C. Lemaréchal, F. Oustry, and C. Sagastizábal. The $\mathcal{U}$-Lagrangian of a convex function. *Transactions of the AMS*, 352(2):711–729, 2000.
232. C. Lemaréchal, F. Pellegrino, A. Renaud, and C. Sagastizábal. Bundle methods applied to the unit-commitment problem. In J. Doležal and J. Fidler, editors, *System Modelling and Optimization*, pages 395–402. Chapman and Hall, 1996.
233. C. Lemaréchal and C. Sagastizábal. An approach to variable metric bundle methods. In J. Henry and J-P. Yvon, editors, *Systems Modelling and Optimization*, number 197 in Lecture Notes in Control and Information Sciences, pages 144–162. Springer Verlag, 1994.
234. C. Lemaréchal and C. Sagastizábal. More than first-order developments of convex functions: primal-dual relations. *Journal of Convex Analysis*, 3(2):1–14, 1996.
235. C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.
236. C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76(3):393–410, 1997.
237. K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.
238. D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–520, 1989.
239. D.G. Luenberger. *Introduction to Linear and Nonlinear Programming* (second edition). Addison-Wesley, Reading, USA, 1984.
240. L. Lukšan and J. Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Math. Programming*, 83(3, Ser. A):373–391, 1998.
241. L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl.*, 102(3):593–613, 1999.
242. Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, 1996.
243. Y. Maday, D. Meiron, A.T. Patera, and E.M. Ronquist. Analysis of iterative methods for the steady and unsteady Stokes problem: application to spectral element discretizations. *SIAM Journal on Scientific Computing*, 14:310–337, 1993.
244. J.H. Maddocks. Restricted quadratic forms, inertia theorems, and the Schur complement. *Linear Algebra and its Applications*, 108:1–36, 1988.
245. Ph. Mahey, S. Oualibouch, and Pham Dinh Tao. Proximal decomposition on the graph of a maximal monotone operator. *SIAM J. Optim.*, 5(2):454–466, 1995.

246. O.L. Mangasarian and S. Fromovitz. The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, 17:37–47, 1967.
247. N. Maratos. *Exact penalty function algorithms for finite dimensional and control optimization problems*. PhD thesis, Imperial College, London, 1978.
248. D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Indust. Appl. Math.*, 11:431–441, 1963.
249. Mathworks. The Matlab distributed computing engine. `http://www.mathworks.com/`.
250. D.Q. Mayne and E. Polak. A superlinearly convergent algorithm for constrained optimization problems. *Mathematical Programming Study*, 16:45–61, 1982.
251. K.A. McShane. A superlinearly convergent $O(\sqrt{n}L)$ iteration interior-point algorithms for linear programming and the monotone linear complementarity problem. *SIAM J. Optimization*, 4:247–261, 1994.
252. R. Mifflin. An algorithm for constrained optimization with semi-smooth functions. *Mathematics of Operations Research*, 2(2):191–207, 1977.
253. R. Mifflin. A modification and an extension of Lemaréchal's algorithm for nonsmooth minimization. *Mathematical Programming Study*, 17:77–90, 1982.
254. R. Mifflin. A quasi-second-order proximal bundle algorithm. *Mathematical Programming*, 73(1):51–72, 1996.
255. R. Mifflin and C. Sagastizábal. Proximal points are on the fast track. *Journal of Convex Analysis*, 9(2):563–579, 2002.
256. R. Mifflin and C. Sagastizábal. Primal-Dual Gradient Structured Functions: second-order results; links to epi-derivatives and partly smooth functions. *SIAM Journal on Optimization*, 13(4):1174–1194, 2003.
257. R. Mifflin and C. Sagastizábal. A $\mathcal{V}U$-proximal point algorithm for minimization. *Math. Program.*, 104(2–3):583–608, 2005.
258. R. Mifflin, D.F. Sun, and L.Q. Qi. Quasi-Newton bundle-type methods for nondifferentiable convex optimization. *SIAM Journal on Optimization*, 8(2):583–603, 1998.
259. S. Mizuno. A new polynomial time method for a linear complementarity problem. *Mathematical Programming*, 56:31–43, 1992.
260. S. Mizuno. A superlinearly convergent infeasible-interior-point algorithm for geometrical LCPs without a strictly complementary condition. *Mathematics of Operations Research*, 21:382–400, 1996.
261. S. Mizuno, F. Jarre, and J. Stoer. A unified approach to infeasible-interior-point algorithms via geometrical linear complementarity problems. *J. Applied Mathematics and Optimization*, 33:315–341, 1994.
262. S. Mizuno, M.J. Todd, and Y. Ye. On adaptative step primal-dual interior-point algorithms for linear programming. *Mathematics of Operations Research*, 18:964–981, 1993.
263. R.D.C. Monteiro and T. Tsuchiya. Limiting behavior of the derivatives of certain trajectories associated with a monotone horizontal linear complementarity problem. *Mathematics of Operations Research*, 21:793–814, 1996.
264. J.J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming, the State of the Art*, pages 258–287. Springer-Verlag, Berlin, 1983.

265. J.J. Moré and D.J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software*, 20:286–307, 1994.
266. J.J. Moré and G. Toraldo. Algorithms for bound constrained quadratic programming problems. *Numerische Mathematik*, 55:377–400, 1989.
267. J.J. Moré and S.J. Wright, editors. *Optimization Software Guide*, volume 14 of *Frontiers in Applied Mathematics*. SIAM Publications, 1993.
268. J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.
269. W. Murray and M. L. Overton. Steplength algorithms for minimizing a class of nondifferentiable functions. *Computing*, 23(4):309–331, 1979.
270. W. Murray and F.J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM Journal on Optimization*, 5:590–640, 1995.
271. W. Murray and M.H. Wright. Projected Lagrangian methods based on the trajectories of penalty and barrier functions. Technical Report SOL-78-23, Department of Operations Research, Stanford University, Stanford, CA 94305, 1978.
272. A.S. Nemirovskii and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics, 1983. (Original Russian: Nauka, 1979).
273. Y.E. Nesterov and A.S. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Number 13 in SIAM Studies in Applied Mathematics. SIAM, Philadelphia, 1994.
274. J. Von Neumann and O. Morgenstein. *Theory of games and economic behavior*. Princeton University Press, Princeton, 1944.
275. J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.
276. J. Nocedal and M.L. Overton. Projected Hessian updating algorithms for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, 22:821–850, 1985.
277. J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 1999.
278. J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, 1970.
279. A. Ouorou. Epsilon-proximal decomposition method. *Math. Program.*, 99(1, Ser. A):89–108, 2004.
280. U.M. Garcia Palomares and O.L. Mangasarian. Superlinear convergent quasi-Newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, 11:1–13, 1976.
281. E.R. Panier and A.L. Tits. Avoiding the Maratos effect by means of a non-monotone line search I: General constrained problems. *SIAM Journal on Numerical Analysis*, 28:1183–1195, 1991.
282. E.R. Panier and A.L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, 59(2):261–276, 1993.
283. E.R. Panier, A.L. Tits, and J. Herskovits. A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization*, 26:788–811, 1988.
284. T. Pietrzykowski. An exact potential method for constrained maxima. *SIAM Journal on Numerical Analysis*, 6:299–304, 1969.

285. E. Polak. *Optimization – Algorithms and Consistent Approximations.* Number 124 in Applied Mathematical Sciences. Springer, 1997.
286. B.T. Polyak. *Introduction to Optimization.* Optimization Software, New York, 1987.
287. F.A. Potra. An $O(nL)$ infeasible-interior-point algorithm for LCP with quadratic convergence. *Annals of Operations Research*, 62:81–102, 1996.
288. M.J.D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, *Optimization*, pages 283–298. Academic Press, London, New York, 1969.
289. M.J.D. Powell. On the convergence of the variable metric algorithm. *Journal of the Institute of Mathematics and its Applications*, 7:21–36, 1971.
290. M.J.D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In R.W. Cottle and C.E. Lemke, editors, *Nonlinear Programming*, number 9 in SIAM-AMS Proceedings. American Mathematical Society, Providence, RI, 1976.
291. M.J.D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14:224–248, 1978.
292. M.J.D. Powell. The convergence of variable metric methods for nonlinearly constrained optimization calculations. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming 3*, pages 27–63, 1978.
293. M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculations. In G.A. Watson, editor, *Numerical Analysis Dundee 1977*, number 630 in Lecture Notes in Mathematics, pages 144–157. Springer-Verlag, Berlin, 1978.
294. M.J.D. Powell. Nonconvex minimization calculations and the conjugate gradient method. In *Lecture Notes in Mathematics 1066*, pages 122–141. Springer-Verlag, Berlin, 1984.
295. M.J.D. Powell. Convergence properties of algorithms for nonlinear optimization. *SIAM Review*, 28:487–500, 1985.
296. M.J.D. Powell. On the quadratic programming algorithm of Goldfarb and Idnani. *Mathematical Programming Study*, 25:46–61, 1985.
297. M.J.D. Powell. The performance of two subroutines for constrained optimization on some difficult test problems. In P.T. Boggs, R.H. Byrd, and R.B. Schnabel, editors, *Numerical Optimization 1984*, pages 160–177. SIAM Publication, Philadelphia, 1985.
298. M.J.D. Powell. A view of nonlinear optimization. In J.K. Lenstra, A.H.G. Rinnooy Kan, and A. Schrijver, editors, *History of Mathematical Programming, A Collection of Personal Reminiscences*, pages 119–125. CWI North-Holland, Amsterdam, 1991.
299. M.J.D. Powell and Y. Yuan. A recursive quadratic programming algorithm that uses differentiable exact penalty functions. *Mathematical Programming*, 35:265–278, 1986.
300. B.N. Pshenichnyj. Algorithm for a general mathematical programming problem. *Kibernetika*, 5:120–125, 1970.
301. B.N. Pshenichnyj. *The Linearization Method for Constrained Optimization.* Number 22 in Computational Mathematics. Springer-Verlag, 1994.
302. B.N. Pshenichnyj and Yu.M. Danilin. *Numerical Methods for Extremal Problems.* MIR, Moscow, 1978.
303. W. Queck. The convergence factor of preconditioned algorithms of the Arrow-Hurwicz type. *SIAM Journal on Numerical Analysis*, 26:1016–1030, 1989.

304. J.K. Reid. A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases. *Mathematical Programming*, 24:55–69, 1982.
305. P.A. Rey and C.A. Sagastizábal. Dynamical adjustment of the prox-parameter in variable metric bundle methods. *Optimization*, 51(2):423–447, 2002.
306. S.M. Robinson. A quadratically convergent algorithm for general nonlinear programming problems. *Mathematical Programming*, 3:145–156, 1972.
307. S.M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Mathematical Programming*, 7:1–16, 1974.
308. S.M. Robinson. Generalized equations and their solutions, part II: applications to nonlinear programming. *Mathematical Programming Study*, 19:200–221, 1982.
309. R.T. Rockafellar. *Convex Analysis*. Number 28 in Princeton Mathematics Ser. Princeton University Press, Princeton, New Jersey, 1970.
310. R.T. Rockafellar. New applications of duality in convex programming. In *Proceedings of the 4th Conference of Probability, Brasov, Romania*, pages 73–81, 1971.
311. R.T. Rockafellar. Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, 12:268–285, 1974.
312. R.T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.
313. R.T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.
314. R.T. Rockafellar and R.J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, 1998.
315. T. Rusten and R. Winthier. A preconditioned iterative method for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 13:887–904, 1992.
316. A. Ruszczyński. Decomposition methods in stochastic programming. In *Mathematical Programming*, volume 79, 1997.
317. C. Sagastizábal and M.V. Solodov. On the relation between bundle methods for maximal monotone inclusions and hybrid proximal point algorithms. In *Inherently parallel algorithms in feasibility and optimization and their applications (Haifa, 2000)*, volume 8 of *Stud. Comput. Math.*, pages 441–455. North-Holland, Amsterdam, 2001.
318. C. Sagastizábal and M.V. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, 16(1):146–169, 2005.
319. R. Saigal. *Linear Programming: A Modern Integrated Analysis*. Kluwer Academic Publishers, Boston, 1995.
320. R.W.H. Sargent. The development of SQP algorithm for nonlinear programming. In L.T. Biegler, T.F. Coleman, A.R. Conn, and F.N. Santosa, editors, *Large-Scale Optimization with Applications, part II: Optimal design and Control*, pages 1–19. IMA Vol. Math. Appl. 93, 1997.
321. R.W.H. Sargent and M. Ding. A new SQP algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 11:716–747, 2000.
322. K. Schittkowski. The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function, Part 1: convergence analysis. *Numerische Mathematik*, 38:83–114, 1981.

323. K. Schittkowski. NLPQL: a FORTRAN subroutine solving constrained nonlinear programming problems. *Annals of Operations Research*, 5:485–500, 1985.
324. K. Schittkowski. Solving nonlinear programming problems with very many constraints. *Optimization*, 25:179–196, 1992.
325. K. Schittkowski. *Numerical Data Fitting in Dynamical Systems*. Kluwer Academic Press, Dordrecht, 2002.
326. H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.
327. Scilab. A free scientific software package. `http://www.scilab.org/`.
328. D.F. Shanno. Conjugate gradient methods with inexact searches. *Mathematics of Operations Research*, 3:244–256, 1978.
329. D.F. Shanno and K.H. Phua. Algorithm 500, minimization of unconstrained multivariate functions. *ACM Transactions on Mathematical Software*, 2:87–94, 1976.
330. A. Shapiro and J. Sun. Some properties of the augmented Lagrangian in cone constrained optimization. *Mathematics of Operations Research*, 29:479–491, 2004.
331. N. Shor. Utilization of the operation of space dilatation in the minimization of convex functions. *Kibernetica*, 1:6–12, 1970. (English translation: *Cybernetics*, 6, 7-15).
332. N.Z. Shor. *Minimization methods for non-differentiable functions*. Springer Verlag, Berlin, 1985.
333. D. Silvester and A. Wathen. Fast iterative solution of stabilized Stokes systems part II: using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31:1352–1367, 1994.
334. M. Slater. Lagrange multipliers revisited: a contribution to non-linear programming. Cowles Commission Discussion Paper, Math. 403, 1950.
335. M. V. Solodov. A class of decomposition methods for convex optimization and monotone variational inclusions via the hybrid inexact proximal point framework. *Optim. Methods Softw.*, 19(5):557–575, 2004.
336. M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Anal.*, 7(4):323–345, 1999.
337. M. V. Solodov and B. F. Svaiter. A hybrid projection-proximal point algorithm. *J. Convex Anal.*, 6(1):59–70, 1999.
338. M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22:1013–1035, 2001.
339. B. Speelpenning. *Compiling fast partial derivatives of functions given by algorithms*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, 1980.
340. P. Spellucci. *Numerische Verfahren der nichtlinearen Optimierung*. Birkhäuser, 1993.
341. P. Spellucci. A new technique for inconsistent problems in the SQP method. *Mathematical Methods of Operations Research*, 47:355–400, 1998.
342. P. Spellucci. An SQP method for general nonlinear programs using only equality constrained subproblems. *Mathematical Programming*, 82:413–448, 1998.
343. J. E. Spingarn. Partial inverse of a monotone operator. *Appl. Math. Optim.*, 10(3):247–265, 1983.

344. M. Spivak. *A Comprehensive Introduction to Differential Geometry*. Publish or Perish, 1979.
345. T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20:626–637, 1983.
346. R.A. Tapia. Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *Journal of Optimization Theory and Applications*, 22:135–194, 1977.
347. R.A. Tapia. On secant updates for use in general constrained optimization. *Mathematics of Computation*, 51:181–202, 1988.
348. T. Terlaky, editor. *Interior Point Methods of Mathematical Programming*. Kluwer Academic Publishers, Boston, 1996.
349. T. Terlaky, J. P. Vial, and K. Roos. *Theory and algorithms for linear optimization: an interior point approach*. Wiley intersciences, New York, 1997.
350. M.J. Todd. On convergence properties of algorithms for unconstrained minimization. *IMA Journal of Numerical Analysis*, 9(3):435–441, 1989.
351. K. Tone. Revisions of constraint approximations in the successive QP method for nonlinear programming problems. *Mathematical Programming*, 26:144–152, 1983.
352. P. Tseng. Alternating projection-proximal methods for convex programming and variational inequalities. *SIAM J. Optim.*, 7(4):951–965, 1997.
353. R.J. Vanderbei. *Linear Programming: Foundations and extensions*. Kluwer Academic Publishers, Boston, 1997.
354. S.A. Vavasis. *Nonlinear Optimization – Complexity Issues*. Oxford University Press, New York, 1991.
355. R. Verfürth. A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem. *IMA Journal of Numerical Analysis*, 4:441–455, 1984.
356. K. Veselić. Finite catenary and the method of Lagrange. *SIAM Review*, 37:224–229, 1995.
357. M. Wagner and M.J. Todd. Least-change quasi-Newton updates for equality-constrained optimization. *Mathematical Programming*, 87:317–350, 2000.
358. A. Wathen and D. Silvester. Fast iterative solution of stabilized Stokes systems part I: using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis*, 30:630–649, 1993.
359. R.B. Wilson. *A simplicial algorithm for concave programming*. PhD thesis, Graduate School of Business Administration, Harvard University, Cambridge, MA, USA, 1963.
360. P. Wolfe. A duality theorem for nonlinear programming. *Quarterly Applied Mathematics*, 19:239–244, 1961.
361. P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11:226–235, 1969.
362. P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical Programming Study*, 3:145–173, 1975.
363. H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming – Theory, Algorithms, and Applications*. Kluwer Academic Publishers, 2000.
364. S.J. Wright. A path-following infeasible-interior-point algorithm for linear complementarity problems. *Optimization Methods and Software*, 2:79–106, 1993.

365. S.J. Wright. An infeasible interior point algorithm for linear complementarity problems. *Mathematical Programming*, 67:29–52, 1994.
366. S.J. Wright. *Primal-dual interior-point methods*. SIAM, Philadelphia, 1996.
367. S.J. Wright. Superlinear convergence of a stabilized SQP method to a degenerate solution. *Computational Optimization and Applications*, 11:253–275, 1998.
368. Y. Xie and R.H. Byrd. Practical update criteria for reduced Hessian SQP: global analysis. *SIAM Journal on Optimization*, 9:578–604, 1999.
369. X. Xu, P.F. Hung, and Y. Ye. A simplification of the homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 62:151–172, 1996.
370. Y. Ye. *Interior point algorithms*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1997.
371. Y. Ye. On homogeneous and self-dual algorithm for LCP. *Mathematical Programming*, 76:211–222, 1997.
372. Y. Ye, M.J. Todd, and S. Mizuno. An $O(\sqrt{n}L)$-iteration homogeneous and self-dual linear programming algorithm. *Mathematics of Operations Research*, 19:53–67, 1994.
373. Y. Yuan. An only 2-step Q-superlinearly convergence example for some algorithms that use reduced Hessian informations. *Mathematical Programming*, 32:224–231, 1985.
374. W.I. Zangwill. Non-linear programming via penalty functions. *Management Science*, 13:344–358, 1967.
375. Y. Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM J. Optimization*, 4:208–227, 1994.
376. G. Zoutendijk. Nonlinear programming, computational methods. In J. Abadie, editor, *Integer and Nonlinear Programming*, pages 37–86. North-Holland, Amsterdam, 1970.