

Luigi Grippo · Marco Sciandrone

Introduction to Methods for Nonlinear Optimization

UNITEXT

La Matematica per il 3+2

Volume 152

Editor-in-Chief

Alfio Quarteroni, Politecnico di Milano, Milan, Italy
École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Series Editors

Luigi Ambrosio, Scuola Normale Superiore, Pisa, Italy
Paolo Biscari, Politecnico di Milano, Milan, Italy
Ciro Ciliberto, Università di Roma “Tor Vergata”, Rome, Italy
Camillo De Lellis, Institute for Advanced Study, Princeton, New Jersey, USA
Victor Panaretos, Institute of Mathematics, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
Lorenzo Rosasco, DIBRIS, Università degli Studi di Genova, Genova, Italy
Center for Brains Mind and Machines, Massachusetts Institute of Technology,
Cambridge, Massachusetts, US
Istituto Italiano di Tecnologia, Genova, Italy

The **UNITEXT - La Matematica per il 3+2** series is designed for undergraduate and graduate academic courses, and also includes advanced textbooks at a research level.

Originally released in Italian, the series now publishes textbooks in English addressed to students in mathematics worldwide.

Some of the most successful books in the series have evolved through several editions, adapting to the evolution of teaching curricula.

Submissions must include at least 3 sample chapters, a table of contents, and a preface outlining the aims and scope of the book, how the book fits in with the current literature, and which courses the book is suitable for.

For any further information, please contact the Editor at Springer:
francesca.bonadei@springer.com

THE SERIES IS INDEXED IN SCOPUS

UNITEXT is glad to announce a new series of free webinars and interviews handled by the Board members, who rotate in order to interview top experts in their field.

Access this link to subscribe to the events:

<https://cassyni.com/events/TPQ2UgkCbJvvz5QbkcWXo3>

Luigi Grippo • Marco Sciandrone

Introduction to Methods for Nonlinear Optimization



Springer

Luigi Grippo
Rome, Italy

Marco Sciandrone
Department of Computer, Control and
Management Engineering
Sapienza University of Rome
Rome, Italy

ISSN 2038-5714
UNITEXT
ISSN 2038-5722
La Matematica per il 3+2
ISBN 978-3-031-26789-5
<https://doi.org/10.1007/978-3-031-26790-1>

ISSN 2532-3318 (electronic)
ISSN 2038-5757 (electronic)
ISBN 978-3-031-26790-1 (eBook)

© Springer Nature Switzerland AG 2023

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Optimization (or mathematical programming) concerns the study of “decision problems”, where the aim is to determine the minimum or the maximum points of a real function (the objective) in a prefixed set (the feasible set). Many real problems arising in the fields of economics, engineering, chemistry, physics, statistics, computer science, management science, operations research and mathematics itself can be formulated as optimization problems.

This book is an introduction to the important area of nonlinear optimization, known also as nonlinear programming, that studies optimization problems where some of the problem functions can be nonlinear. In particular, we restrict our study to continuous nonlinear optimization problems, defined on finite dimensional real spaces, under the assumption that the problem functions are smooth. Even with these restrictions, nonlinear programming has many relevant applications in the solution of nonlinear systems of equations and inequalities, in the construction and the analysis of mathematical models, in the solution of (almost all) engineering problems, in the computation of the control laws of discrete-time dynamical systems. At present, nonlinear optimization techniques represent also a basic ingredient of *machine learning* methods, for the computation of the parameters of the learning system on the basis of the available data.

The book contains three blocks of chapters concerning:

- Basic theory and optimality conditions (Chaps. 1–7)
- Unconstrained and constrained algorithms (Chaps. 8–23)
- Advanced topics (Chaps. 24–26).

The first two blocks consist of short chapters, and each of these chapters may correspond, in the authors’ experience, to the notes for one or two lessons (with each lesson of approximately 2 hours). Some more space is given to the matter of the third block. In each chapter, we add a few exercises and a terminal section on notes and references. As the main emphasis of the book is on solution methods, we believe that the best instructive exercise could be the realization of a computer code, related to the methods illustrated in the book, under the guidance of the teacher.

The basic concepts and results of linear algebra, of calculus on R^n and of convex analysis, are reported in Chaps. 27, 28, and 29. In particular, in Chap. 29 on convex analysis, we give also the proof of all the results used in the book, since, in many cases, this chapter must be included in a course. Then, the only prerequisite is the content of basic courses on linear algebra and calculus.

Most of the basic theoretical results reported in the text are formally proved in order to allow the students, potential readers of the book, to acquire a sound background in optimization. As regards “classical” optimization subjects, we attempted to give the simplest proofs available in the literature or adapted from the best known sources. Particular attention is devoted to the study of global convergence properties of the algorithms, since the definition of globalization techniques represents one of the major contributions of optimization to computational methods.

The book contains also more specialized topics, which are receiving an increasing interest in the last years, like those concerning derivative-free methods, non-monotone methods, decomposition algorithms, which have been, and currently are, research topics of the authors, and are not usually reported in introductory textbooks.

The material given in the book can be used for constructing different courses, both at an (advanced) undergraduate level and at a graduate level.

A short course (say of 30 hours) can be based on elements of convex analysis (Chap. 29), basic definitions and existence results (Chap. 2), optimality conditions (at least Chaps. 3, 4, and 5). This material can be combined, for instance, with a basic course on linear programming. Note, however, that linear programming is not a prerequisite and that the basic theory of linear programming can also be derived, as special case, from the optimality conditions of nonlinear programming.

A course of 60 hours for advanced undergraduate students could be based, for instance, on the same material indicated above, with the addition of a selection of unconstrained algorithms that includes at least Chaps. 8, 9, 10, and 11.

Similarly, a course of 60 hours for graduated students could be focused on a selection of classes of methods presented in the text, both for unconstrained and constrained optimization problems. The choice of the arguments will depend on the type of master degree and on the background and scientific interests of the potential readers of the book.

Some important topics like, for instance, sparse optimization, incremental and online algorithms, complexity analysis, non-smooth optimization, global optimization, cone programming, semidefinite programming have been leaved out for space limitations. We believe, however, that the study of the material reported in the book can be a useful prerequisite for further extensions. The basic techniques introduced here can also be useful in the study of other sectors of decision sciences like variational inequalities, multi objective optimization and game theory.

Finally we remark that most of the material on unconstrained methods is extracted from the book (in Italian):

L. Grippo and M. Sciandrone, *Metodi di ottimizzazione non vincolata*, Springer-Verlag Italia, Milano, 2011, which contains also some of the proofs omitted here.

Contents

1	Introduction	1
1.1	Optimization Problems	1
1.2	Classification of Optimization Problems	3
1.3	Continuous Optimization Problems Over R^n	4
1.3.1	Classification	4
1.3.2	Examples of Continuous Problems.....	6
1.4	Appendix: Examples of Problems in Different Spaces.....	13
1.4.1	An Infinite-Dimensional Optimization Problem: Control of a Dynamical System	14
1.4.2	An Integer Linear Programming Problem: Scheduling of Sport Tournaments	14
1.4.3	A Mixed Integer Programming Problem: Clustering	17
1.5	Notes and References	19
2	Fundamental Definitions and Basic Existence Results	21
2.1	Local and Global Minimum Points	21
2.2	Minimizers in Convex Programming	24
2.3	Minimizers in Generalized Convex Problems	26
2.4	Minimizers in Concave Programming	29
2.5	Equivalent Transformations	30
2.5.1	Equivalence Criterion	30
2.5.2	Some Basic Examples.....	31
2.5.3	Fractional Programming	33
2.5.4	Goal Programming	34
2.6	Existence Conditions	36
2.6.1	Existence in Unconstrained Optimization.....	37
2.6.2	Existence in Constrained Optimization	44
2.6.3	Feasibility Problems	47
2.7	Exercises	49
2.8	Notes and References	51

3	Optimality Conditions for Unconstrained Problems in R^n	53
3.1	Descent Directions	53
3.2	Optimality Conditions	56
3.3	Optimality Conditions in the Convex Case	60
3.4	Exercises	66
3.5	Notes and References	67
4	Optimality Conditions for Problems with Convex Feasible Set	69
4.1	Optimality Conditions Based on Feasible Directions	69
4.2	The Case of Feasible Set Defined by Linear Constraints	71
4.2.1	Feasible Directions for Problems with Box Constraints	74
4.3	Optimality Conditions over a Convex Set	76
4.4	Projection on a Convex Set	80
4.5	Exercises	86
4.6	Notes and References	86
5	Optimality Conditions for Nonlinear Programming	87
5.1	Problem Formulation and Notation	87
5.2	Fritz John Conditions	88
5.3	Constraint Qualifications and KKT Conditions	93
5.4	Sufficient Conditions in the Convex Case	100
5.5	Second Order Optimality Conditions	101
5.6	Linearly Constrained Problems	109
5.6.1	Problems with Non Negativity Constraints	110
5.6.2	Problems with Box Constraints	111
5.6.3	Problems with Simplex Constraints	112
5.6.4	Quadratic Programming	113
5.6.5	Linear Programming	114
5.7	Exercises	115
5.8	Notes and References	117
6	Duality Theory	119
6.1	The Dual Problem	119
6.2	Lagrangian Duality	121
6.2.1	Basic Concepts, Weak and Strong Lagrangian Duality	121
6.2.2	Saddle Point Optimality Conditions	125
6.3	Wolfe's Dual	126
6.3.1	Wolfe's Dual in Quadratic Convex Programming Problems with Linear Inequality Constraints	130
6.4	Appendix	132
6.5	Exercises	134
6.6	Notes and References	135

7	Optimality Conditions Based on Theorems of the Alternative	137
7.1	Introduction	137
7.2	Optimality Conditions for Problems with Inequality Constraints	138
7.3	Optimality Conditions Based on Tangent Directions	141
7.4	Optimality Conditions for Problems with Equality and Inequality Constraints	142
7.5	Fritz John Conditions Established with Motzkin Theorem	147
7.5.1	KKT Conditions from FJ Conditions	148
7.6	Appendix	150
7.6.1	Linear Inequalities and Theorems of the Alternative	150
7.6.2	Non Homogeneous Farkas Lemma and Optimality for LP	162
7.7	Notes and References	165
8	Basic Concepts on Optimization Algorithms	167
8.1	Structure of Optimization Algorithms	167
8.2	Existence and Uniqueness of Limit Points	169
8.3	Convergence Towards Critical Points	170
8.4	Local and Global Convergence	171
8.5	Convergence Rate	172
8.6	Basic Concepts of Complexity Analysis	174
8.7	Notes and References	175
9	Unconstrained Optimization Algorithms	177
9.1	Preliminary Concepts	177
9.2	Limit Points	178
9.3	Convergence Towards Stationary Points	179
9.4	Classification of Unconstrained Algorithms	180
9.5	Convergence of Line Search Based Methods	181
9.6	Exercises	185
9.7	Notes and References	186
10	Line Search Methods	187
10.1	Basic Features and Classification of Line Searches	187
10.2	Armijo's Method	191
10.3	Step-Length Extension and Goldstein Conditions	198
10.4	Wolfe Conditions	201
10.5	Derivative-Free Line Searches	204
10.5.1	Backtracking Armijo-Type Derivative-Free Algorithms	205
10.5.2	Derivative-Free Linesearches with Step Extension	208
10.6	Appendix A: Algorithms Employing Wolfe Conditions	213
10.7	Appendix B: Implementation of Line Searches	216
10.7.1	Initial Interval	217
10.7.2	Initial Estimate of the Step-Size	217

10.7.3	Interpolation Formulas	219
10.7.4	Application of Interpolation Formulas	224
10.7.5	Stopping Criteria and Failures	225
10.8	Exercises	226
10.9	Notes and References	227
11	Gradient Method	229
11.1	The Steepest Descent Direction	229
11.2	The Gradient Method	230
11.3	Gradient Method with Constant Step-Size	232
11.4	Convergence Rate	234
11.5	Finite Convergence in the Quadratic Case	238
11.6	Complexity of the Steepest Descent Method	239
11.7	Modified Gradient Methods	243
11.8	Exercises	246
11.9	Notes and References	247
12	Conjugate Direction Methods	249
12.1	The Conjugate Direction Method	249
12.2	The Conjugate Gradient Method (CGM): The Quadratic Case	252
12.3	Convergence Rate	258
12.4	Preconditioning	259
12.5	The CGM in the Non Quadratic Case	262
12.6	Fletcher-Reeves Method	265
12.7	Method of Polyak-Polak-Ribière (PPR)	266
12.8	Appendix: The CG Method When the Hessian Matrix is Positive Semidefinite	271
12.9	Exercises	273
12.10	Notes and References	273
13	Newton's Method	275
13.1	The Pure Newton Iteration	275
13.2	Local Convergence and Convergence Rate	277
13.3	Shamanskii Method	280
13.4	Globalization of Newton's Method for Minimization	283
13.5	Hybrid Methods	289
13.6	Modification Methods	291
13.7	Exercises	294
13.8	Notes and References	295
14	Trust Region Methods	297
14.1	The General Trust Region Framework and Convergence Results	297
14.2	Classification of Solution Methods for the Trust Region Subproblem	301
14.3	The Cauchy Step-Based Method	302
14.4	The Dogleg Method	304

14.5	The Conjugate Gradient Method of Steihaug	306
14.6	Necessary and Sufficient Optimality Conditions for the Trust Region Subproblem	308
14.7	Trust Region Approach to Globalizing Newton's Method	311
14.8	Complexity Issues and Adaptive Regularized Methods	316
14.9	Appendix: Proofs of Convergence	318
14.10	Exercises	323
14.11	Notes and References	324
15	Quasi-Newton Methods	325
15.1	Preliminaries	325
15.2	Rank 1 Formulae	328
15.3	Rank Two Formulae and the BFGS Method	329
15.4	Global Convergence of the BFGS Method	334
15.5	Convergence Rate	335
15.6	Exercises	336
15.7	Notes and References	336
16	Methods for Nonlinear Equations	337
16.1	Preliminaries	337
16.2	Newton-Type Methods	339
16.3	Broyden's Method	342
16.4	Residual Based Methods	344
16.5	Notes and References	344
17	Methods for Least Squares Problems	345
17.1	Least Squares Problems	345
17.2	Linear Least Squares Problems	347
17.3	Methods for Nonlinear Least Squares Problems	349
17.4	Gauss-Newton Method	350
17.5	Levenberg-Marquardt Method	355
17.6	Recursive Linear Least Squares Algorithm	358
17.7	Some Notes on Incremental Methods for Nonlinear Problems ...	360
17.8	Exercises	362
17.9	Notes and References	363
18	Methods for Large-Scale Optimization	365
18.1	Solution Strategies for Large Scale Problems	365
18.2	Truncated Newton Method	366
18.3	Globally Convergent Truncated Newton Methods	367
18.3.1	Preliminaries	367
18.3.2	A Truncated Newton Method Based on a Line Search	368
18.4	Quasi-Newton Methods for Large-Scale Optimization	375
18.4.1	Preliminaries	375
18.4.2	Memoryless Quasi-Newton Methods	376
18.4.3	Limited-Memory Quasi-Newton Methods	376

18.5	Exercises	380
18.6	Notes and References	380
19	Derivative-Free Methods for Unconstrained Optimization	383
19.1	Motivation and Classification of Derivative-Free Methods	383
19.2	The Coordinate Method	385
19.2.1	Preliminary Concepts	385
19.2.2	Coordinate Method with Simple Decrease	386
19.3	The Hooke-Jeeves Method	392
19.4	The Nelder-Mead Method	394
19.5	Linesearch-Based Methods	396
19.5.1	Basic Assumptions and Convergence Conditions	398
19.5.2	Globalization of Direct Search Methods Through Line Searches	403
19.6	Approximation of Derivatives and Implicit Filtering	405
19.6.1	Simplex Gradient	405
19.6.2	Implicit Filtering	408
19.6.3	Combination with Coordinate Search	409
19.7	Model-Based Methods	409
19.8	Exercises	411
19.9	Notes and References	411
20	Methods for Problems with Convex Feasible Set	413
20.1	Problems with Convex Feasible Set	413
20.2	Line Search Along a Feasible Direction	415
20.3	The Frank-Wolfe Method (Conditional Gradient Method)	419
20.4	Gradient Projection Method	421
20.5	A Derivative-Free Method for Box Constrained Problems	425
20.6	Concave Programming for Minimizing the Zero-Norm Over Polyhedral Sets	434
20.7	Frank-Wolfe Method Applied to Concave Programming Problems	437
20.8	Exercises	439
20.9	Notes and References	439
21	Penalty and Augmented Lagrangian Methods	441
21.1	Basic Concepts	441
21.2	Sequential Penalty Functions	444
21.3	Augmented Lagrangian Functions	452
21.4	Non Differentiable Exact Penalty Functions	459
21.5	Continuously Differentiable Exact Penalty Functions	459
21.6	Exact Shifted Barrier Functions	462
21.7	Exact Augmented Lagrangian Functions	471
21.8	Exercises	478
21.9	Notes and References	478

22	SQP Methods	481
22.1	Newton Type Methods for the Equality Constrained Case	481
22.2	Extension to Inequality Constrained Problems	484
22.3	EQP Methods	485
22.4	Quasi-Newton Modification	488
22.5	Solution of the Quadratic Programming Subproblem	489
22.6	Globalization Strategies	490
22.6.1	Nondifferentiable Merit Functions	491
22.6.2	Smooth Merit Functions	494
22.7	Notes and References	496
23	Introduction to Interior Point Methods	497
23.1	Basic Definitions and Barrier Methods	497
23.2	Interior Point Methods for Linear Programming	503
23.2.1	Definitions and Notation	503
23.2.2	A Summary of Basic LP Theory	504
23.2.3	Main Classes of IPMs for LP and Basic Assumptions	505
23.2.4	Feasible Path-Following Methods	509
23.2.5	Infeasible Methods	519
23.3	Extensions	521
23.3.1	Quadratic Programming	521
23.3.2	IPMs for Nonconvex Problems	522
23.4	Appendix: Regularity and Optimality	524
23.5	Notes and References	526
24	Nonmonotone Methods	529
24.1	Motivation and Basic Concepts	529
24.2	Convergence Issues in Nonmonotone Methods	531
24.3	Reference Values and Preliminary Results	532
24.4	Armijo-Type Nonmonotone Line Searches	538
24.5	Nonmonotone Armijo-Goldstein and Parabolic Searches	541
24.6	Nonmonotone Derivative-Free Linesearches	546
24.7	Watchdog Techniques	552
24.8	Nonmonotone Globalization of Newton's Method	557
24.9	Nonmonotone Inexact Newton-Type Methods for Nonlinear Equations	560
24.10	Notes and References	570
25	Spectral Gradient Methods	573
25.1	The BB Method in the Quadratic Case	573
25.2	Nonmonotone Globalization of the BB Method	578
25.3	Spectral Gradient Methods for Nonlinear Equations	580
25.4	Projected Spectral Gradient Method for Minimization on Convex Sets	585

25.5	Exercises	586
25.6	Notes and References	586
26	Decomposition Methods	589
26.1	Motivation and Examples	589
26.2	Classes of Decomposition Methods and Basic Notation	592
26.3	Block Gauss-Seidel (GS) Method and Extensions	595
26.3.1	The Basic Scheme of the GS Method	595
26.3.2	A Line Search Algorithm in the Component Space	596
26.3.3	Limit Points of the GS Algorithm	599
26.3.4	The Two-Block GS Method	601
26.3.5	Convergence of the GS Method Under Generalized Convexity Assumptions	602
26.3.6	Proximal-Point Modification of the GS Method	607
26.4	Block Descent Methods	609
26.4.1	A Basic Unconstrained Block Descent Algorithm	609
26.4.2	A Basic Constrained Block Descent Algorithm	612
26.5	The Gauss-Southwell Algorithm	615
26.6	Decomposition with Variable and Overlapping Blocks	619
26.7	The Jacobi Method	624
26.8	Algorithms for Problems with a Single Linear Equality Constraint and Box Constraints	626
26.9	Sequential Minimal Optimization (SMO) Algorithms	627
26.10	Appendix: Proof of Proposition 26.16	633
26.11	Exercises	637
26.12	Notes and References	637
27	Basic Concepts of Linear Algebra and Analysis	639
27.1	The Space R^n as Linear Space	639
27.2	Matrices and Systems of Linear Equalities	641
27.3	Norm, Metric, Topology, Scalar Product Over R^n	644
27.4	Notation and Results on Real Matrices	651
27.4.1	Determinant, Minors, Trace	651
27.4.2	Norm of a Matrix	652
27.4.3	Orthogonal Matrices	654
27.4.4	Eigenvalues of Symmetric Matrices	654
27.4.5	Singular Value Decomposition	656
27.5	Quadratic Forms	658
28	Differentiation in R^n	663
28.1	First Order Derivatives of a Real Function	663
28.2	Differentiation of a Vector Function	665
28.3	Second Order Derivatives of a Real Function	667
28.4	The Mean Value Theorem and Taylor's Theorem	669
28.5	Derivatives of Composite Functions	671
28.6	Examples	672

29	Introduction to Convex Analysis	681
29.1	Convex Sets	681
29.2	Convex Functions	689
29.3	Composition of Convex Functions	692
29.4	Convexity of Differentiable Functions	695
29.5	Monotonicity Conditions on ∇f	700
29.6	Basic Notions of Generalized Convexity	703
References		709
Index		719

Chapter 1

Introduction



1.1 Optimization Problems

The mathematical representation of an optimization problem, starting from a logical, informal and qualitative description of a decision problem, requires to associate suitable *variables*, i.e. the *unknowns* of the mathematical representation, to the quantities involved in the decision problem. The definition of the variables and, in particular, the definition of the space X of the variables, strongly influences the methods of analysis and the choice of the solution techniques to be adopted.

The relationships that must be satisfied among the variables and the limitations deriving, for instance, from “physical” considerations, must be quantitatively expressed. These relationships and limitations define the *constraints*: the set of values of the variables satisfying the constraints is the *feasible set* $S \subseteq X$.

If the problem is related to a “real” context and is well-formulated we would expect that the feasible set is not empty. Typically, it contains a very large number (possibly infinite) of elements. Then an *objective function* to be minimized or maximized over the feasible set can be defined as a real-valued function $f : S \rightarrow R$.

Summarizing, the optimization problems considered in the book will take the form:

$$\min_{x \in S} f(x)$$

where $S \subseteq X$ is the feasible set and $f : S \rightarrow R$ is the objective function. As $\max f(x), x \in S$ has the same solutions of $\min (-f(x)), x \in S$, in the analysis of the optimization problem we will assume, without loss of generality, that f has to be minimized.

Below we consider a simple optimization problem.

Example 1.1 One of the most studied models in the context of Microeconomics tries to capture the behaviour of the consumer. In the simplest case we assume that the consumer can choose the goods in a set of n elements and in a prefixed temporal interval. We denote by $x_j \in R$, for $j = 1, \dots, n$, the quantity of the j -th good consumed in the given period and by $p_j \in R$ the unit cost of the j -th good. We can define the constraints

$$\sum_{j=1}^n p_j x_j \leq M \quad x_j \geq 0, \quad j = 1, \dots, n,$$

where M is the budget of the consumer. The first constraint, known as *budget constraint*, states that the expense for purchasing goods can not exceed the available budget. The *nonnegativity constraints* on x_j are immediate consequence of the definition of the variables. We assume that the goods are infinitely divisible and hence that the space of variables is R^n .

Setting $x = (x_1, \dots, x_n)^T$ and $p = (p_1, \dots, p_n)^T$, the feasible set can be formally defined as

$$S = \{x \in R^n : p^T x \leq M, \quad x \geq 0\}.$$

Assume we can define over S a real-valued function $U : S \rightarrow R$, called *utility function*, that provides the “value” of any choice of the consumer. Then we can introduce the *optimization problem* of determining, among all the vectors $x \in S$, the one where the utility function takes the maximum value. The existence of an utility function can be ensured under suitable axioms on the order of the *preferences* of the consumer [176]. \square

Starting from the mathematical model of an optimization problem it is possible to analytically deduce some important properties, on the basis of general results established for the class of models that includes our problem.

The theoretical issues of a given class of optimization problems may concern:

- existence and uniqueness of the optimal solution;
- analytical characterization of the optimal solutions via the optimality conditions (necessary, sufficient, necessary and sufficient);
- stability of the optimal solutions with respect to the model parameters and/or the data used to build the model.

In this book we will confine ourselves to consider only the essential theoretical results at the basis of the solution methods we will describe. In particular, in the general case (when the feasible set is non empty), we are interested in the characterization of the points that satisfy the necessary optimality conditions, which we will define as *critical points* of the problem.

In most of cases the “solution” (in some well defined sense) of an optimization problem can be obtained or approximated only using a computational algorithm. Therefore, many *optimization algorithms* have been designed and realized to solve

the most important classes of optimization problems. There exist several software libraries containing the codes of the most popular and reliable methods, which are widely used in many different application fields.

However, the adoption of a numerical procedure may involve several issues related to the choice of the method to be employed. This choice may be not unique since, for a given class of problems, there exist, in general, several available computational methods. Moreover, the behaviour of a method may strongly depend on the values of the parameters characterizing the algorithm. In order to solve a complex problem, it can be important to exploit a good knowledge both of the structure of the problem and of the properties of the available algorithms. In some difficult cases it could be necessary also to modify appropriately some standard code.

An introduction to the study of the best known optimization algorithms for nonlinear optimization is one of the main objectives of this book.

1.2 Classification of Optimization Problems

Optimization problems can be classified in several ways taking into account the space of variables, the structure of the feasible set and the properties of the involved functions.

A preliminary, basic classification considers:

- *infinite-dimensional* optimization problems where usually the solution must be sought in a space of functions, whose elements can be vectors of functions of one or more variables;
- *discrete* optimization problems, where the variables can assume only integer values;
- *finite-dimensional continuous* optimization problems, where the space of variables is a finite-dimensional linear space, typically R^n .

The class of infinite-dimensional problems includes, in particular, *calculus of variations*, *optimal control* problems, *optimal approximation* of functions.

Discrete optimization concerns in particular:

- *integer programming* problems, where the space of variables is the set Z^n of n -tuples of integer numbers;
- *combinatorial optimization* problems, where the space of variables is $\{0, 1\}^n$, that is, the set of vectors with n components in $\{0, 1\}$;
- *network optimization problems*, where the objective function and the constraints are defined on interconnected systems often represented by a graph.

Between continuous and discrete optimization there is *mixed integer programming*, which includes problems where only some variables are constrained to take integer values.

We report in the appendix of the chapter examples of problems belonging to the above classes. In the next section we will focus on the class of continuous optimization problems that represents the topic of the book.

1.3 Continuous Optimization Problems Over R^n

1.3.1 Classification

In this book we will consider only continuous optimization problems whose feasible set is contained in R^n . These problems can be further classified taking into account the structure of the feasible set S and the assumptions on the objective function f .

If $S = R^n$ then the problem is an *unconstrained optimization* problem. If S is a proper subset of R^n then the problem is a *constrained optimization* problem.

Unconstrained optimization problems may include, in particular the problem of solving a set of nonlinear equations or inequalities defined on R^n .

In fact, suppose we have the system

$$h(x) = 0,$$

where $h : R^n \rightarrow R^p$ is a vector of given functions. For solving this system we can consider the unconstrained problem

$$\min f(x) \equiv \|h(x)\|^q, \quad x \in R^n,$$

where $\|\cdot\|$ is a given norm and $q > 0$. It is easily seen that a solution of the system is also a solution of the optimization problem. Conversely, if x^* solves the optimization problem and we have $f(x^*) = 0$, it solves also the system. Thus there are strong connections between the methods of Numerical Analysis and the optimization methods. The specific contribution of Optimization is that of providing techniques that extend the convergence region of nonlinear methods, such as, for instance Newton's method.

A similar technique can also be employed for solving a system of nonlinear inequalities, such as

$$g(x) \leq 0,$$

where $g : R^n \rightarrow R^m$ is a given vector of functions. In this case we can define the functions

$$g_i^+(x) = \max\{0, g_i(x)\}, \quad i = 1, \dots, m$$

and the vector

$$g^+(x) = \begin{pmatrix} g_1^+(x) \\ \vdots \\ g_m^+(x) \end{pmatrix}$$

and consider the unconstrained problem

$$\min f(x) \equiv \|g^+(x)\|^q, \quad x \in R^n,$$

where $q > 0$. In fact, we have $\|g^+(x^*)\|^q = 0$ if and only if $g(x^*) \leq 0$.

Among the constrained optimization problems, the most common class is represented by problems whose feasible set is formally described by a finite set of *equality and inequality constraints*, that is

$$S = \{x \in R^n : g(x) \leq 0, h(x) = 0\},$$

where $g : R^n \rightarrow R^m$ and $h : R^n \rightarrow R^p$ are vectors of given functions.

There is also an important class of problems whose feasible set is defined by an infinite number of constraints. These problems belong to the class of *semi-infinite programming* problems. This class of problems includes, for instance, optimal approximation of functions and problems characterized by the presence of a constraint requiring that a certain matrix must be positive semidefinite, i.e., *semidefinite programming* problems.

Further classifications can be introduced taking into account the properties of the objective function and of the constraints, such as the *linearity*, the *differentiability*, the *convexity*.

With reference to the *hypothesis of linearity* we can distinguish between:

- *linear programming* problems, where the objective function is linear and the feasible set is defined by a system of linear equalities and inequalities; and
- *nonlinear programming* problems, where the objective function is nonlinear and/or at least one constraint is defined by a nonlinear function.

Taking into account the properties of differentiability, we can distinguish between:

- *differentiable* optimization problems, where the functions defining the objective and the constraints are continuously differentiable functions over the feasible set; and
- *nondifferentiable* optimization problems (which are necessarily nonlinear problems), where the above condition is not satisfied.

In terms of convexity properties, we can consider:

- *convex programming* problems, concerning minimization problems whose feasible set is convex and the objective function is convex over it;
- *nonconvex programming* problems, where the above conditions is not satisfied.

Examples of continuous problems arising in quite different contexts are illustrated in the next paragraph.

1.3.2 Examples of Continuous Problems

Some examples of optimization problems defined on R^n are reported here.

1.3.2.1 A Linear Programming Problem: Transportation Problem

Consider a company producing and selling a commodity. The company has a set of N sources, where the commodity is available, and a set of M demand centers where the commodity is sold.

At each source s_i , with $i \in \{1, \dots, N\}$, a given quantity q_i of commodity is available. Each demand center o_j , with $j \in \{1, \dots, M\}$ requires a given quantity d_j . We indicate by c_{ij} the cost per unit of transporting the commodity from source s_i to demand center o_j .

The problem is to determine the quantity to be transported from each source to each demand center in such a way that:

- the capacity constraints of the sources are satisfied;
- the requirements of the demand centers are satisfied;
- the total transportation cost is minimized.

To formulate the problem, we indicate by x_{ij} the quantity of the commodity transported from source s_i to demand center o_j . Then the optimization problem is

$$\begin{aligned} \min_x & \sum_{i=1}^N \sum_{j=1}^M c_{ij} x_{ij} \\ \sum_{j=1}^M x_{ij} & \leq q_i \quad \text{for all } i \in \{1, \dots, N\} \\ \sum_{i=1}^N x_{ij} & = d_j \quad \text{for all } j \in \{1, \dots, M\} \\ x_{ij} & \geq 0 \quad i = 1, \dots, N, j = 1, \dots, M. \end{aligned}$$

1.3.2.2 An Unconstrained Optimization Problem: Neural Network Training

An interesting unconstrained optimization problem is that concerning the *supervised training of artificial neural networks*, which are mathematical models of learning systems inspired by the structure and functioning of the neuronal system of advanced organisms.

In this class of models the elementary computational unit is the *neuron* (or *formal neuron*) that performs a transformation of an *input vector* u into a scalar *output* $y(u)$.

In the simplest structure the components of the input are multiplied by (positive or negative) *weights*, representing the strength of the synaptic connections, so that we obtain a weighted sum of the inputs. The difference between this algebraic sum and a given *threshold* value is transformed by an *activation function* g (usually nonlinear) into the output y .

If we use as activation function the *sign function*:

$$g(t) = \text{sgn}(t) = \begin{cases} 1 & t \geq 0 \\ -1 & t < 0. \end{cases}$$

the neuron provides the output 1 if the weighted sum of the inputs is greater than the threshold value, and the output -1 otherwise. A scheme of the formal neuron with $g(t) = \text{sgn}(t)$ is shown in Fig. 1.1.

The limitations of a *single adaptive layer* of formal neurons motivated the study of architectures defined by a cascade of two-or more layers of neurons. These architectures are known as *multilayer feed-forward neural networks*, where there exists at least one *hidden layer*, that is a layer between the input layer and the output layer. In the feed-forward neural networks the activation function g of every neuron is usually differentiable.

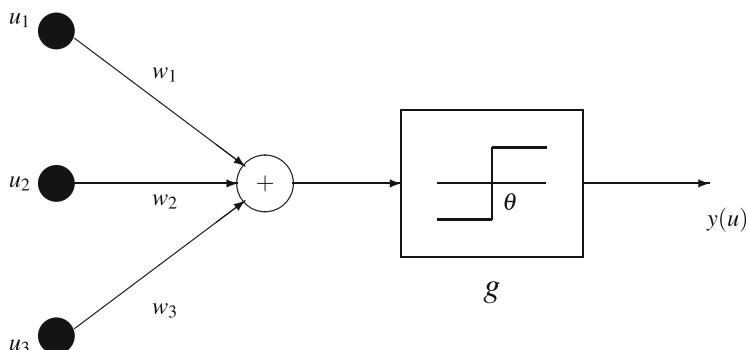


Fig. 1.1 Scheme of a formal neuron

The most common activation functions are the *logistic function*

$$g(t) = \frac{1}{1 + e^{-ct}}, \quad c > 0$$

that yields the output in $(0, 1)$ and the *hyperbolic tangent function*

$$g(t) = \tanh(t/2) = \frac{1 - e^{-t}}{1 + e^{-t}},$$

that provides the output in $(-1, 1)$.

Let us consider a neural network with m inputs, one hidden layer with N neurons having activation function g , and one output neuron whose activation function is a linear function.

We adopt the following notation:

- w_{ji} : the weights of the connections between the input nodes and the hidden layer neurons;
- θ_j : the threshold of the hidden neuron j ;
- v_j : the weights of the connections between the hidden layer neurons and the output neuron;
- g : activation function of the hidden neurons.

The scheme with a two-layer neural network, with $m = 2$ and $N = 3$ is shown in Fig. 1.2.

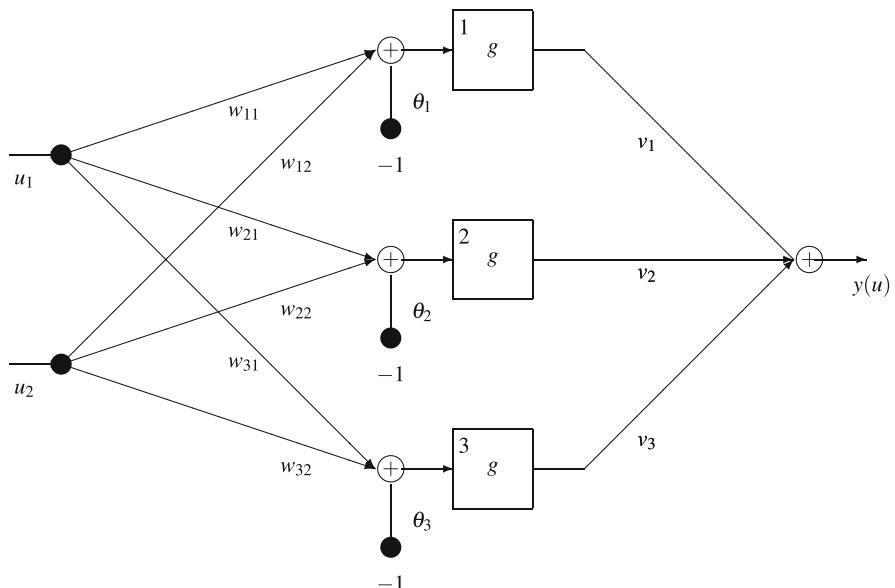


Fig. 1.2 Neural network with 2 layers, with 1 hidden layer, 2 inputs, 1 output

The output of a neural network having an input vector $u \in R^m$ and one hidden layer of N neurons becomes:

$$y(w; u) = \sum_{j=1}^N v_j g \left(\sum_{i=1}^m w_{ji} u_i - \theta_j \right),$$

where the parameter vector $w \in R^n$, with $n = (m + 1)N + N$ is

$$w = (w_{11}, \dots, w_{1N}, w_{21}, \dots, w_{2N}, \dots, w_{m1}, \dots, w_{mN}, \theta_1, \dots, \theta_N, v_1, \dots, v_N)^T.$$

The computation of the parameters by the training process, starting from the set of input-output data

$$T = \{(u^p, y^p), u^p \in R^m, y^p \in R, p = 1, \dots, P\},$$

leads to the following unconstrained optimization problem

$$\min E(w) = \sum_{p=1}^P E_p(w), \quad w \in R^n,$$

where E_p is the error term related to the p -th sample and measures the distance between the desired output (label) y^p and the actual output $y(w; u^p)$ of the network. The most common measure is the quadratic error

$$E_p(w) = \frac{1}{2}(y(w; u^p) - y^p)^2.$$

1.3.2.3 A Nonlinear Optimization Problem with Box Constraints: Design of a Compact Low Field Magnetic Resonance Imaging Magnet

Magnetic resonance imaging (MRI) is one of the most useful tools for clinical diagnosis and biomedical research. It is performed with a large instrument that allows the patient to be inserted into a region of uniform magnetic field. The field is generated either by an electromagnet (resistive or superconductive) or by a permanent magnet.

In [237] a “dedicated” smaller apparatus based on resistive magnets has been designed. The field is generated by four symmetric pairs of coils: I_1, I_2, I_3, I_4 . The required field value is B_0 and must be as uniform as possible in a spherical region Ω at the center of the magnet. Let x, y, z be the axes of a Cartesian tern having the origin at the center of the magnet. A physical point r_j is represented by the coordinates (x_j, y_j, z_j) . The vector of currents is denoted by $I = (I_1, I_2, I_3, I_4)$ and the z component of the magnetic field generated in a point r_j by $B_z(I; r_j)$. The

problem that we want to solve through the optimization is to find a vector of currents I^* such that

$$\frac{|B_z(I^*; r_j) - B_0|}{B_0} \leq \epsilon \quad \text{for all } r_j \in \Omega$$

where ϵ is a sufficiently small positive number. Given a point $r_j \in \Omega$ the function

$$B_z(., r_j) : R^4 \rightarrow R$$

is not known analytically, but for each $I \in R^4$ the value $B_z(I; r_j)$ can be obtained experimentally.

Consider a sufficient number N_p of uniformly distributed points $r_j \in \Omega$ and define the *black-box* optimization problem

$$\min_I \sum_{j=1}^{N_p} (B_z(I; r_j) - B_0)^2$$

$$0 \leq I \leq U,$$

where the lower bounds ($0 \leq I$) are imposed because the currents values must have the same sign for reasons related to the construction of the magnet. The upper bounds ($I \leq U$) depend on the allowed power dissipation.

1.3.2.4 A Finite-Dimensional Continuous Optimization Problem: Portfolio Selection Problem

We present the portfolio selection problem as example of continuous optimization problem. Portfolio selection theory studies how to allocate an investor's available capital into a prefixed set of assets with the aims of maximizing the expected return and minimizing the investment risk. Let n be the number of available assets, let $\mu \in R^n$ be the vector of expected returns of the assets, and let $Q \in R^{n \times n}$ be the symmetric positive semidefinite matrix whose generic element q_{ij} is the covariance of returns of assets i and j .

We assume that the available (normalized) capital is fully invested. Then, let $x \in R^n$ be the vector of decision variables, where x_i is the fraction of the available capital to be invested into asset i , with $i = 1, \dots, n$. Then, vector x must satisfy the constraints

$$e^T x = 1$$

$$x \geq 0,$$

where $e \in R^n$ denotes the column vector of all ones. By this notation, $\mu^T x$ is the expected return of the portfolio and $x^T Qx$ is the covariance of the portfolio which can be used as a measure of the risk connected with the investment. In the traditional Markowitz portfolio selection model [182], the optimization problem is stated as the following convex quadratic programming problem

$$\min_x \frac{1}{2} x^T Qx$$

$$\mu^T x = \beta$$

$$e^T x = 1$$

$$x \geq 0,$$

where β is the desired expected return of the portfolio.

1.3.2.5 A Convex Optimization Problem with Linear Constraints: Traffic Assignment Problem

Traffic assignment concerns the problem of forecasting the loadings on a transportation network where users choose routes from their origins to their destinations.

Assignment can be performed using one of the two alternative principles enunciated by Wardrop [98]:

- the *user-optimal principle*, which leads to a network equilibrium where the travel costs of all the routes actually used are equal to or less than those on unused routes;
- the *system-optimized principle*, stating that the choices of the users are such that a system aggregate criterion is optimized.

We focus here on the traffic assignment problem based on the user-optimal principle. The network is modelled by a direct graph, whose nodes represent origins, destinations, and intersections, and arcs represent the transportation links. There is a set of node pairs, called Origin/Destination (OD).

Let $G = (N, A)$ be the network graph with N as the set of nodes and A as the set of arcs. Let $P \subseteq N \times N$ be the set of all Origin/Destination (O/D) pairs for which is defined a travel demand $D_p > 0$, $D_p \in R$. With K_p we indicate the finite set of paths connecting an O/D pair $p \in P$, in which only a-cyclic (simple) paths are considered. The set of all paths is indicated with K ,

$$K = \cup_{p \in P} K_p \tag{1.1}$$

For convenience, let $n_p = |K_p|$ and $n = |K|$. We denote by $x \in R^n$ the vector of path flows and by x_k the flow on path $k \in K$. For each O/D pair $p \in P$ we define its

block of path flows variables with $x_{(p)} \in R^{n_p}$ and its i -th element as $x_{(p),i}$. Through the definition of an arc-path incidence matrix such as

$$\delta_{ak} = \begin{cases} 1 & \text{if arc } a \in A \text{ belongs to path } k \in K \\ 0 & \text{otherwise} \end{cases} \quad (1.2)$$

for all $a \in A$ and $k \in K$, the arc flow v_a is a function of the vector of path flows x and it is obtained as follows

$$v_a(x) = \sum_{k \in K} \delta_{ak} x_k, \quad \text{for all } a \in A. \quad (1.3)$$

Each arc $a \in A$ of the network has associated a cost function $s_a(\cdot)$ which, in general, may be dependent on the overall flow distribution in the network. We assume:

- *separability* of the costs, i.e., $s_a(\cdot)$ is a function only of the flow $v_a(x)$ of the arc $a \in A$;
- *additivity* of the path cost functions, i.e., the cost $s_k(\cdot)$ of a path $k \in K$ is

$$s_k(x) = \sum_{a \in A} \delta_{ak} s_a(v_a(x)); \quad (1.4)$$

- *strictly increasing costs*, i.e., the arc cost $s_a(\cdot)$, for all $a \in A$, is a continuously differentiable and strictly increasing function.

Under the above assumptions the Traffic Assignment Problem (TAP) can be formulated as the following convex programming problem [98]:

$$\begin{aligned} \min_x \quad & \sum_{a \in A} \int_0^{\sum_{k \in K} \delta_{ak} x_k} s_a(t) dt \\ & \sum_{i \in \{1, \dots, n_p\}} x_{(p),i} = D_p, \quad \text{for all } p \in P \\ & x_{(p)} \geq 0 \quad \text{for all } p \in P \end{aligned} \quad (1.5)$$

We observe that the convexity of the objective function follows from the assumption of strictly increasing costs.

1.3.2.6 A Constrained Optimization Problem with Nonlinear Constraints: Novelty Detection

During recent years, many efforts have been devoted to the development of *unsupervised* learning tools introduced in order to solve pattern recognition problems where

samples coming only from one class (*normal samples*) are available. This leads to the so-called *novelty detection* problem [248], which can be described as follows.

Let $TS = \{x^1, \dots, x^\ell\}$ be the training set made of *normal* vectors $x^i \in R^n$, $i = 1, \dots, \ell$, and assume that the training vectors have been generated by a source whose outputs are vectors belonging to a set $\Omega \subset R^n$ which is not a-priori known. Given any new pattern $x \in R^n$, the task is that of establishing whether x is *normal* ($x \in \Omega$) or is *abnormal* ($x \notin \Omega$).

Assume that the unknown set $\Omega \subset R^n$ is an hypersphere. Then, a simple approach for novelty-detection is that of determining the smallest hypersphere containing “most” of the training vectors x^i , for $i = 1, \dots, \ell$, and of establishing that new patterns are novel whenever they lie outside the hypersphere so defined. This leads to the following non smooth unconstrained problem

$$\min_{c \in R^n, r \in R} r^2 + C \sum_{i=1}^{\ell} \max\{0, \|x^i - c\|^2 - r^2\}$$

or equivalently, by introducing additional variables ξ_i , to the following smooth constrained problem

$$\begin{aligned} & \min_{c \in R^n, r \in R, \xi \in R^\ell} r^2 + C \sum_{i=1}^{\ell} \xi_i \\ & \|x^i - c\|^2 \leq r^2 + \xi_i \quad i = 1, \dots, \ell, \\ & \xi_i \geq 0 \quad i = 1, \dots, \ell, \end{aligned} \tag{1.6}$$

where the parameter $C > 0$ permits to control the trade-off between the size of the radius r and the number of training patterns outside the hypersphere, and hence to reduce the effect of possible outliers.

Once determined the elements characterizing the hypersphere, i.e., the centre \bar{c} and the radius \bar{r} , the learning model gives a boundary around the target data set. This boundary is used to decide whether new objects are target objects or outliers.

1.4 Appendix: Examples of Problems in Different Spaces

We give now some examples of optimization problems in the main classes defined in the Sect. 1.2.

1.4.1 An Infinite-Dimensional Optimization Problem: Control of a Dynamical System

A classical example of infinite-dimensional problem is the optimal control problem, where the task is to determine the “best” control law for a dynamical system in such a way that a given functional is optimized. We consider here a “simple” example.

Given a point of mass m following a rectilinear uniform motion along the real line, let u be the scalar function of the time representing the force acting on it in the fixed interval $[0, t_f]$. The problem is to determine the control law u in such a way that the point at the final instant t_f is close to the origin of the real line by minimizing, as much is possible, the control energy. Let $x(t)$, $v(t)$ be position and velocity of the point at a generic instant t , and let \bar{x}_0 and \bar{v}_0 be their given initial values, i.e., $x(0) = \bar{x}_0$, $v(0) = \bar{v}_0$. From the laws of classical physics we get that the dynamical system is defined by the following time-continuous differential equations

$$\begin{aligned} v(t) &= \frac{dx(t)}{dt} & x(0) &= \bar{x}_0 \\ u(t) &= m \frac{dv(t)}{dt} & v(0) &= \bar{v}_0 \end{aligned}$$

Then the problem is to determine a function u , on the interval $[0, t_f]$, to minimize the cost functional

$$J(u) = c_1 x(t_f)^2 + c_2 \int_0^{t_f} u(t)^2 dt,$$

with $c_1, c_2 > 0$.

This problem can be formulated, for instance, as a constrained problem where the variables are the functions u , x , v defined on $[0, t_f]$ and belonging to appropriate function spaces. The choice of the space of variables in many cases is not a simple problem, since we must guarantee that the dynamical system has a solution and that the optimization problem is solvable. We refer, for instance, to [206] for a discussion of these aspects. The constraints are represented by the dynamical equations that define the motion.

1.4.2 An Integer Linear Programming Problem: Scheduling of Sport Tournaments

As example of discrete optimization problem, we consider an integer linear programming problem that concerns the scheduling of sport tournaments [49]. Sports scheduling in professional sports usually requires the definition of dates and venues

of matches between teams attending a tournament, and represents an interesting application field of optimization methodologies.

The schedule of a tournament should satisfy requirements that can be imposed based on tournament structure, television rights, and marketing revenues. Here we consider a simplified version of a *Double Round-Robin* (DRR) structure, where each team plays exactly twice with each other team, once in each half. The second half of the DRR is a mirror of the first half, with home games and away games exchanged.

Consider a tournament with n teams (for simplicity we assume that n is an even number). The games must be associated with the $n - 1$ slots for each half. Every team has its venue in its hometown in which the team must play exactly one of the two matches played against each other team. When a team plays at its venue, it plays a *home* game; at any other venue, it plays an *away* game. A *home-away* pattern is the sequence of home games and away games played by a team during the tournament. Two consecutive home games or away games are defined as a *break*. The basic constraints are the following:

- each team must play exactly one game in a single slot;
- each team must meet all other teams twice, one time in the first half of the tournament, and one time in the other half;
- consecutive breaks are forbidden;
- there are specific slots in which a team i is forced to play a home game, or a team h must play an away game in those slots.

Travel issues must be considered for slots played in particular days (e.g., games scheduled on December 26). The objective could be that of minimizing the total distance traveled, for instance, in the slot of December 26.

We have the binary decision variables defined as follows

$$\delta_{ij}^k = \begin{cases} 1 & \text{if team } i \text{ plays at home against team } j \text{ in slot } k \\ 0 & \text{otherwise.} \end{cases}$$

All the variables and the constraints relate to the first half of a round-robin tournament; we obtain the second half by mirroring the first half. To formulate the constraints concerning consecutive breaks we need auxiliary variables. In particular, HH_i^k is a binary variable that states if team i plays at home in both slot k and $k + 1$; the variable AA_i^k for away breaks is defined in a similar way. The relationships

$$\sum_{j \neq i} (\delta_{ij}^k + \delta_{ij}^{k+1}) = 2 \Rightarrow HH_i^k = 1$$

$$\sum_{j \neq i} (\delta_{ji}^k + \delta_{ji}^{k+1}) = 2 \Rightarrow AA_i^k = 1$$

can be modeled as follows

$$\sum_{j \neq i} (\delta_{ij}^k + \delta_{ij}^{k+1}) \leq 1 + HH_i^k$$

$$\sum_{j \neq i} (\delta_{ji}^k + \delta_{ji}^{k+1}) \leq 1 + AA_i^k$$

Then, the constraints are the following:

- each team must play exactly one game in a slot, i.e.,

$$\sum_{j \neq i} \delta_{ij}^k + \sum_{j \neq i} \delta_{ji}^k = 1 \quad \text{for all } k, i;$$

- each team meets all other teams once in each half, i.e.,

$$\sum_k \delta_{ij}^k + \sum_k \delta_{ji}^k = 1 \quad \text{for all } i, j, i \neq j;$$

- no two consecutive breaks are allowed, i.e.,

$$HH_i^k + HH_i^{k+1} \leq 1 \quad \text{for all } i, k$$

$$AA_i^k + AA_i^{k+1} \leq 1 \quad \text{for all } i, k$$

- a team h must play at home in a slot k_h and a team p must play away in a slot k_p , i.e.,

$$\sum_{j \neq h} \delta_{hj}^{k_h} = 1$$

$$\sum_{j \neq p} \delta_{jp}^{k_p} = 1$$

Finally, the objective function to be minimized, that is, the total distance traveled in the slot \bar{k} (December 26), is

$$\sum_i \sum_{j \neq i} d_{ij} \delta_{ij}^{\bar{k}},$$

being d_{ij} the distance between the home towns of teams i and j .

The whole *integer linear programming* problem is reported below

$$\begin{aligned}
 & \min_{\delta} \sum_i \sum_{j \neq i} d_{ij} \delta_{ij}^k \\
 & \sum_{j \neq i} \delta_{ij}^k + \sum_{j \neq i} \delta_{ji}^k = 1 \quad \text{for all } i, k \\
 & \sum_k \delta_{ij}^k + \sum_k \delta_{ji}^k = 1 \quad \text{for all } i, j, i \neq j \\
 & \sum_{j \neq i} \left(\delta_{ij}^k + \delta_{ij}^{k+1} \right) \leq 1 + HH_i^k \quad \text{for all } i \\
 & \sum_{j \neq i} \left(\delta_{ji}^k + \delta_{ji}^{k+1} \right) \leq 1 + AA_i^k \quad \text{for all } i \\
 & HH_i^k + HH_i^{k+1} \leq 1 \quad \text{for all } i, k \\
 & AA_i^k + AA_i^{k+1} \leq 1 \quad \text{for all } i, k \\
 & \sum_{j \neq h} \delta_{hj}^{k_h} = 1 \\
 & \sum_{j \neq p} \delta_{jp}^{k_p} = 1 \\
 & \delta_{ij}^k \in \{0, 1\} \quad \text{for all } i, j, k \\
 & HH_i^k, AA_i^k \in \{0, 1\} \quad \text{for all } i, k
 \end{aligned}$$

1.4.3 A Mixed Integer Programming Problem: Clustering

As example of problem where we have both continuous and discrete variables, we present the clustering problem [32].

Assume we have a set of *unlabeled* data (i.e., vectors in R^n) that must be put in similarity groups called *clusters*. A cluster is a collection of patterns which are “similar” between them, and “dissimilar” to patterns in other clusters.

The clustering problem can be formulated in different ways. Here we consider one of the best known formulations. Let us assume that the set (*training set*) of patterns

$$TS = \{x^i, x^i \in X \subseteq R^n, i = 1, \dots, N\}$$

is given. Let $d(., .) : R^n \times R^n \rightarrow R^+$ be a measure of *dissimilarity* between two vectors in R^n . Given a norm $\|\cdot\|$ in R^n , for any pair of vectors $x, y \in R^n$ we can think to define a measure of dissimilarity $d(x, y)$ between x and y as the *distance*

$$d(x, y) = \|x - y\|.$$

We define the *dissimilarity of a cluster* as the sum of the *dissimilarities* between the patterns of the cluster and a *centroid* (to be determined) of the cluster.

We assume that the number M of clusters is given. We want to partition the training set into M clusters and to define the centroids of the clusters in such a way that the sum of the dissimilarities of the clusters is minimized.

Let us introduce the *binary variables*

$$\delta_{ij} = \begin{cases} 1 & \text{if } x^i \in \text{cluster } j \\ 0 & \text{otherwise} \end{cases}$$

and the vectors of *continuous variables* $z_j \in R^n, j = 1, \dots, M$, each one represents the centroid of the corresponding cluster. By definition, the dissimilarity of cluster j is

$$s_j = \sum_{i=1}^N \delta_{ij} d(x^i, z_j).$$

Each pattern x^i , for $i = 1, \dots, N$ must be assigned to one and only one cluster. This leads to the constraints

$$\sum_{j=1}^M \delta_{ij} = 1 \quad i = 1, \dots, N.$$

Then the clustering problem takes, in principle, the following form of *mixed integer programming* problem

$$\begin{aligned} \min_{\delta, z} & \sum_{i=1}^N \sum_{j=1}^M \delta_{ij} d(x^i, z_j) \\ \sum_{j=1}^M \delta_{ij} &= 1 \quad i = 1, \dots, N \\ \delta_{ij} &\in \{0, 1\} \quad i = 1, \dots, N, j = 1, \dots, M \\ z_j &\in R^n \quad j = 1, \dots, M. \end{aligned} \tag{1.7}$$

Note that by replacing the constraints

$$\delta_{ij} \in \{0, 1\}$$

by the box constraints

$$0 \leq \delta_{ij} \leq 1$$

we obtain a continuous optimization problem. It is possible to prove that this latter problem is “equivalent” to the mixed integer programming problem (1.7).

1.5 Notes and References

We give some essential bibliographic indication on the subjects treated in this book and on extensions to other areas based on mathematical models and optimization algorithms.

As already stated, we restrict our interest to smooth nonlinear continuous optimization methods defined on finite dimensional spaces. Even with these limitations, however, the relevant literature is so ample that we can only indicate the works we will most often refer in the sequel. Our basic references on nonlinear equations and continuous finite dimensional optimization include the books: [16, 179, 196, 200]. Other important references are reported in the notes at the end of Chap. 2.

A minimal set of suggestions for further reading on related areas are the following books. Linear programming: [19, 91] Global optimization: [145, 146, 167, 205]. Optimal control and function space optimization: [3, 45, 115, 149, 168, 253]. Stochastic optimization [234]. Semidefinite programming [258]. Modern convex optimization [14]. Variational inequalities: [87]. Game Theory: [202]. Integer programming and Combinatorial optimization: [118, 191, 235]. Approximation theory: [33, 218]. Model building in mathematical programming: [256]. Neural networks and machine learning: [24, 139, 140, 243]

Chapter 2

Fundamental Definitions and Basic Existence Results



In this chapter we first introduce the basic definitions of global and local solutions of optimization problems on R^n . Then we discuss the characterization of minimizers in some important classes of minimization problems, such as convex programming problems, generalized convex problems, concave programming problems. We introduce also an elementary criterion for defining *equivalent transformations* between different formulations of some classes of problems. Finally, we consider some sufficient existence conditions for unconstrained and constrained problems.

2.1 Local and Global Minimum Points

Let $S \subseteq R^n$ and let $f : S \rightarrow R$ be a real function defined on S . As we have already seen, the optimization problem in R^n can be formulated, without loss of generality, as the problem of determining a minimum point (if it exists) of the *objective function* f in the *feasible set* S , provided that this set is nonempty.

We start with the following fundamental definition.

Definition 2.1 (Global Minimum Point) A point $x^* \in S$ is said to be a *global minimum point* of f on S if

$$f(x^*) \leq f(x), \quad \text{for all } x \in S,$$

and the value $f(x^*)$ is a *global minimum* of f on S .

(continued)

Definition 2.1 (continued)

The point $x^* \in S$ is said to be a *strict global minimum point* of f on S if:

$$f(x^*) < f(x), \quad \text{for all } x \in S, x \neq x^*.$$

□

A minimization problem is typically written in the form

$$\begin{aligned} \min_{x \in S} f(x), \end{aligned}$$

where the symbol “*min*” expresses, informally, the intention of minimizing f , even if the problem has no solution.

The set (possibly empty) of optimal solutions is often indicated with the notation $\operatorname{Arg} \min_{x \in S} f(x)$.

Example 2.1 Consider the problem

$$\begin{aligned} \min_{x \in S} f(x) = x_1^2 - x_2 \\ x \in S \end{aligned}$$

where $S = \{x \in R^2 : x_2 \leq 0\}$. We note that $f(x) \geq 0$ for every $x \in S$. It is easily seen that $x^* = (0 \ 0)^T$ is a (strict) global minimum point, as $x^* \in S$, and $0 = f(x^*) < f(x)$ for all $x \in S, x \neq x^*$. Therefore we have

$$f(x^*) = \min_{x \in S} f(x) \quad \text{and} \quad \{x^*\} = \operatorname{Arg} \min_{x \in S} f(x).$$

If we set $S = \{x \in R^2 : x_2 \geq 0\}$ then the points $x(t) = (0 \ t)^T$ for all $t > 0$ belong to S and we have $f(x(t)) \rightarrow -\infty$ for $t \rightarrow \infty$. Thus does not exist a minimum point and $\operatorname{Arg} \min_{x \in S} f(x) = \emptyset$. □

In many optimization problems, when the search for global minima can be difficult, it is also of interest searching for *local minimizers* of f . To define formally these points we consider, as the *neighborhood* of a given point $x \in R^n$, the open ball $B(x; \rho)$ with center x and radius $\rho > 0$, that is $B(x; \rho) = \{y \in R^n : \|x - y\| < \rho\}$, where $\|\cdot\|$ is a norm on R^n . Thus, we can state the following definition.

Definition 2.2 (Local Minimum Point) A point $x^* \in S$ is said to be a *local minimum point* of f on $S \subseteq R^n$ if there exists a neighborhood $B(x^*; \rho)$ of x^* , with $\rho > 0$ such that

$$f(x^*) \leq f(x), \quad \text{for all } x \in S \cap B(x^*; \rho)$$

and, in this case, we say that $f(x^*)$ is a *local minimum* of f on S .

We say that $x^* \in S$ is a *strict local minimum point* of f on S if there exists a neighborhood $B(x^*; \rho)$ of x^* , with $\rho > 0$ such that

$$f(x^*) < f(x), \quad \text{for all } x \in S \cap B(x^*; \rho), \quad x \neq x^*.$$

□

We have obviously that a global minimizer is also a local minimizer. We also observe that if S has a nonempty interior and there exists $B(x^*; \rho) \subseteq S$, such that

$$f(x^*) \leq f(x), \quad \text{all } x \in B(x^*; \rho),$$

we can consider x^* as a local *unconstrained minimum point* of f on S .

The definitions introduced above are illustrated in Figs. 2.1 and 2.2 with reference to the case where the space of variables is the real axis.

In Fig. 2.1 we suppose that x is unconstrained, and that the feasible set is the real line. We also assume that there are only two minimizers. In this case, \bar{x} is a local strict minimum point, while x^* is the unique global minimizer, which is also strict.

In Fig. 2.2 we suppose that the feasible set is the closed interval $S = [a, b]$. The point a is a local strict minimizer; the interval $(y, z]$, is constituted of local unconstrained minimizers. The unique global minimizer is the point x^* , which is also a strict and unconstrained global minimizer. We can observe that the point of

Fig. 2.1 Unconstrained
minimizers on $S = R$

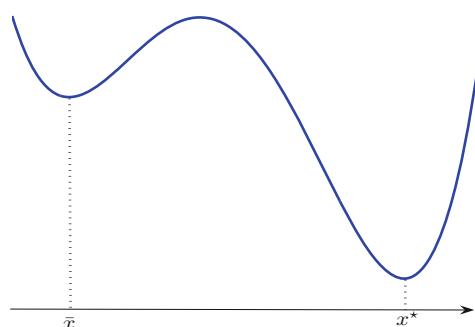
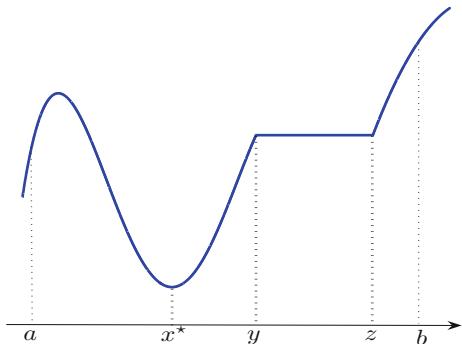


Fig. 2.2 Local and global minimizers and maximizers on $S = [a, b]$



the open interval (y, z) are both local minimizers and local maximizers. Finally, the point y is a local maximizers, the point z is a local minimizer and b is the unique strict global maximizer.

2.2 Minimizers in Convex Programming

Convex programming problems are the *minimization problems* where the feasible set S is a convex set and the objective function $f : S \rightarrow R$ is convex or, equivalently, the *maximization problems*, where the feasible set is convex, but the objective function is concave. In the sequel we will refer to minimization problems, unless otherwise specified.

Concepts and results concerning convexity are reported in Chap. 29. Here we briefly recall the definitions of convex set and convex function.

Definition 2.3 (Convex Set) A set $S \subseteq R^n$ is *convex* if, for every pair $x_1, x_2 \in S$, the line segment $[x_1, x_2]$ is contained in S , that is

$$x_1, x_2 \in S, \quad \lambda \in R, \quad 0 \leq \lambda \leq 1 \quad \text{imply} \quad (1 - \lambda)x_1 + \lambda x_2 \in S.$$

Definition 2.4 (Convex Function) Let $S \subseteq R^n$ be a convex set and let $f : S \rightarrow R$. We say that f is *convex* on S if, for every pair $x, y \in S$ we have

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y),$$

(continued)

Definition 2.4 (continued)

for all $\lambda \in R$ such that $0 \leq \lambda \leq 1$. We say that f is *strictly convex* on S if, for every pair $x, y \in S$ with $x \neq y$ we have

$$f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y),$$

for all λ such that $0 < \lambda < 1$.

In the case of convex programming problems, as shown by the next proposition, local minimizers are also global minimizers.

Proposition 2.1 (Minimizers in the Convex Case) *Let $S \subseteq R^n$ be a convex set and let f be a convex function $f : S \rightarrow R^n$. Then, every local minimum point of f on S is also a global minimum point. Moreover, the set of global minimizers is a convex set.*

Proof Let x^* be a local minimum point of f on S and let $x \neq x^*$ a feasible point. As x^* is a local minimizer there must exist an open ball $B(x^*; \rho)$ with $\rho > 0$ such that

$$f(x^*) \leq f(y), \quad \text{for all } y \in B(x^*; \rho) \cap S$$

and hence, by convexity of S and convexity of $B(x^*; \rho)$, we can find a value λ , with $0 < \lambda \leq 1$, such that the point

$$z = (1 - \lambda)x^* + \lambda x$$

belongs to $B(x^*; \rho) \cap S$, which implies

$$f(x^*) \leq f(z).$$

By convexity of f , we have also

$$f(z) = f((1 - \lambda)x^* + \lambda x) \leq (1 - \lambda)f(x^*) + \lambda f(x).$$

From the preceding inequalities we obtain

$$f(x^*) \leq f(z) \leq (1 - \lambda)f(x^*) + \lambda f(x),$$

whence, as $\lambda > 0$, we obtain $f(x^*) \leq f(x)$, which proves the first assertion.

Let now $X^* \subseteq S$ be the set of global optimal solutions; if we have $X^* = \emptyset$ or $X^* = \{x^*\}$ the assertion is obviously true.

Thus, let x^*, y^* be two points of X^* , so that

$$f(x^*) = f(y^*) = \min_{x \in S} f(x).$$

By the convexity assumptions, we must have, for every $\lambda \in [0, 1]$

$$f((1 - \lambda)x^* + \lambda y^*) \leq (1 - \lambda)f(x^*) + \lambda f(y^*) = f(x^*),$$

which implies that $[x^*, y^*] \subseteq X^*$ and hence that X^* is convex. \square

From Proposition 2.1, by replacing f with $-f$, it follows that

iff f is a function concave on S then every local maximum point off on S is also a global maximum point.

Uniqueness of the minimum point can be established, in particular, if f is strictly convex on S . The next proposition establishes a sufficient (but not necessary) uniqueness condition. The proof is left as an exercise.

Proposition 2.2 (Uniqueness of the Global Minimum : f Strictly Convex)

Let $S \subseteq R^n$ be a convex set and let f be a function strictly convex on S . Then, if x^ is a local minimizer of f on S , the point x^* is also the unique local and global minimum point of f on S .* \square

2.3 Minimizers in Generalized Convex Problems

Some of the results stated in the preceding section can be extended by replacing the convexity hypothesis on f with a generalized convexity hypothesis. First we recall the following definitions (see, e.g. Chap. 29).

Let $S \subseteq R^n$ be a convex set and let $f : S \rightarrow R$. Function f is

- (i) *quasi-convex* on S if, for every pair $x, y \in S$, for every λ such that $0 \leq \lambda \leq 1$, we have:

$$f((1 - \lambda)x + \lambda y) \leq \max\{f(x), f(y)\};$$

- (ii) *strictly quasi-convex* on S if, for every pair $x, y \in S$ such that $f(x) \neq f(y)$, for every λ such that $0 < \lambda < 1$, we have:

$$f((1 - \lambda)x + \lambda y) < \max\{f(x), f(y)\};$$

- (iii) *strongly quasi-convex* on S if for every pair $x, y \in S$ such that $x \neq y$, for every λ such that $0 < \lambda < 1$, we have:

$$f((1 - \lambda)x + \lambda y) < \max\{f(x), f(y)\}.$$

We consider the assumptions which imply that a local minimum point of f over S is also a global minimum point. In this case we can replace the convexity assumption on f , used in Proposition 2.1 with the assumption of *strict quasi-convexity*.

Proposition 2.3 (Minimizers in the Case of Strict Quasi-Convexity) *Let $S \subseteq R^n$ be a convex set and f a strictly quasi-convex function over S . Then every local minimum point of f in S is a global minimum point.*

Proof Let x^* be a local minimum point of f over S . Let us assume, by contradiction, that there exists a point $\hat{x} \in S$ such that $f(\hat{x}) < f(x^*)$. Since x^* is a local minimizer there exists an open ball $B(x^*; \rho)$ with $\rho > 0$ such that

$$f(x^*) \leq f(y), \quad \text{for all } y \in B(x^*; \rho) \cap S.$$

The convexity of S implies that there exists a $\lambda \in (0, 1)$ such that

$$z = (1 - \lambda)x^* + \lambda\hat{x} \in B(x^*; \rho) \cap S;$$

this implies that

$$f(x^*) \leq f(z). \tag{2.1}$$

On the other hand, by the assumption of strict quasi-convexity of f we have:

$$f(z) = f((1 - \lambda)x^* + \lambda\hat{x}) < \max\{f(x^*), f(\hat{x})\} = f(x^*)$$

and this contradicts (2.1). □

Note that the quasi-convexity assumption is not sufficient to ensure that each local minimizer is also a global minimizer. See, for instance Fig. 2.3 where it is shown a quasi-convex function with local non global minimizers.

Fig. 2.3 A quasi-convex function with local, non global minimizers

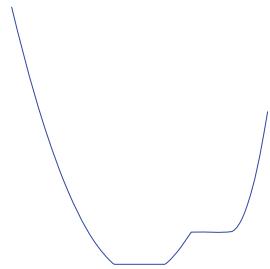
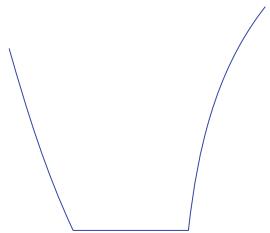


Fig. 2.4 Strictly quasi-convex function: every local minimum is a global minimum



The strict quasi-convexity assumption excludes the presence of local non global minimizers, however it is not sufficient to ensure the uniqueness of the global solution (see e.g. Fig. 2.4).

Uniqueness can be ensured under the hypothesis that the function is strongly quasi-convex. We can state the following proposition.

Proposition 2.4 (Uniqueness of the Global Minimizer: f Strongly Quasi-Convex) Let $S \subseteq R^n$ be a convex set and let f be a strongly quasi-convex function over S . If x^* is a local minimum point of f over S , then x^* is the unique local and global minimum point of f over S .

Proof Let x^* be a local minimizer of f over S . Since the strong quasi-convexity implies the strict quasi-convexity, from the preceding proposition we get that x^* is a global minimizer.

Let us assume, reasoning by contradiction, that there exists a point $\hat{x} \in S$ such that $f(\hat{x}) = f(x^*)$ and $\hat{x} \neq x^*$. Taking into account the convexity of S , there must exist a number λ , with $0 < \lambda < 1$, such that

$$z = (1 - \lambda)x^* + \lambda\hat{x} \in S$$

Then the strong quasi-convexity of f implies:

$$f(z) = f((1 - \lambda)x^* + \lambda\hat{x}) < \max\{f(x^*), f(\hat{x})\} = f(x^*)$$

and this contradicts the fact that x^* is a global minimizer. \square

2.4 Minimizers in Concave Programming

Concave programming problems are those where the feasible set $S \subseteq R^n$ is convex and the objective function $f : S \rightarrow R$ to be minimized is concave.

This important class includes, in particular, linear programming problems, which are both convex and concave problems.

In the general case, concave programming problems can be difficult problems with many local minimizers and constitute some of the most studied problems in global optimization. It can be shown that a large class of integer programming problems can be transformed into concave programming problems. See, e.g. [110].

An interesting result, concerning the location of minimizers, is of great value in the study of concave programming problems.

We can establish the following proposition.

Proposition 2.5 (Minimizers in Concave Programming) *Let $S \subseteq R^n$ be a convex set and let f be a concave function $f : S \rightarrow R^n$. Suppose that f admits a global optimal solution $x^* \in S$ and that f is not constant on S . Then, every global optimal solution is a boundary solution.*

Proof Let $x^* \in S$ be a global minimizer of f in S and set $f^* = f(x^*)$. As f is non constant on S , there must exist $y \in S$ such that $f(y) > f^*$. If $\text{int}(S) = \emptyset$ the assertion holds trivially. Therefore assume that $x \in \text{int}(S)$ is an interior point and consider a ball $B(x; \rho) \subset \text{int}(S)$, with $\rho > 0$. Let $d = y - x$ be a line passing through y and x . Define $z = x - \varepsilon d$ and assume that $\varepsilon \|d\| < \rho$. Then, we can define the segment $[z, y] \subseteq S$, by imposing

$$[z, y] = (1 - \lambda)z + \lambda y, \quad 0 \leq \lambda \leq 1.$$

It can be easily verified that the point x is a point of this segment, for an appropriate value of λ (take $\lambda = \varepsilon/(1 + \varepsilon)$). Thus, by the assumptions made, we have that $f(z) \geq f^*$ and $f(y) > f^*$. Therefore, by the concavity of f , we can write:

$$f(x) = f((1 - \lambda)z + \lambda y) \geq (1 - \lambda)f(z) + \lambda f(y) > f^*.$$

This shows that an interior point can not be an optimal minimizer and hence the assertion holds. \square

2.5 Equivalent Transformations

In many cases, it could be convenient to transform a given problem into an “equivalent” problem with a different structure, such that a solution of the original problem can be more easily obtained from a solution of the transformed problem. In this section we first introduce our definition of “equivalence” and we state a simple criterion for proving equivalence. Then we consider various examples of elementary equivalent transformations; in particular we show that some nonlinear non differentiable problems can be easily transformed into equivalent linear programming problems. Interesting applications of useful transformations are also introduced with reference to *fractional programming* and to *goal programming*.

2.5.1 Equivalence Criterion

Let us consider two problems of the form

$$\min_{x \in S} f(x) \quad (\text{P})$$

$$\min_{y \in D} g(y). \quad (\text{Q})$$

We say that the two problems are *equivalent* if:

- (i) problem (P) has an optimal solution if and only if problem (Q) has an optimal solution;
- (ii) if the problems admit a solution then we can establish a correspondence between the solutions of the two problems in the sense that a solution of one problem can be (easily) obtained from a solution of the other problem.

A simple condition, which establishes points (i) and (ii) above can be the following.

Proposition 2.6 (Equivalent Transformations) *Let $S \subseteq R^n$, $f : S \rightarrow R$, $D \subseteq R^m$ and $g : D \rightarrow R$. Suppose there exist two mappings $\rho : S \rightarrow D$ and $\sigma : D \rightarrow S$, such that at each $x \in S$ we can associate an element $\rho(x) \in D$ and to each $y \in D$ an element $\sigma(y) \in S$, in a way that the following conditions are satisfied.*

- (i) *For every $x \in S$, we have $g(\rho(x)) \leq f(x)$.*
- (ii) *For every $y \in D$, we have: $f(\sigma(y)) \leq g(y)$.*

(continued)

Proposition 2.6 (continued)
Then:

- (a) if x^* is a global minimum point of f on S the point $\rho(x^*) \in D$ is a global minimum point of g on D ;
- (b) if y^* is a global minimum point of g on D the point $\sigma(y^*) \in S$ is a global minimum point of f on S .

Proof Suppose that g has a global minimizer y^* on D . By (ii) there must exist $x^* = \sigma(y^*) \in S$ such that $f(x^*) \leq g(y^*)$. We show that x^* is a global minimizer of f on S . Reasoning by contradiction let us assume that there exists a point $\bar{x} \in S$ such that $f(\bar{x}) < f(x^*)$. By condition (i) we have that there exists a point $\bar{y} = \rho(\bar{x}) \in D$ such that $g(\bar{y}) \leq f(\bar{x})$. As y^* is a global minimizer of g on D , from the preceding inequalities we get

$$g(y^*) \leq g(\bar{y}) \leq f(\bar{x}) < f(x^*) \leq g(y^*),$$

which is a contradiction. By exchanging the role of the two problems, the inverse implication can be proved by repeating the same reasoning. \square

2.5.2 Some Basic Examples

In this section we give some basic examples of equivalent transformations. Consider first the problem (P)

$$\min_{1 \leq i \leq m} \max \{f_i(x)\}, \quad x \in S$$

where the objective function is the maximum of a finite number of functions $f_i : S \rightarrow R$:

$$f(x) = \max_{1 \leq i \leq m} \{f_i(x)\}$$

Let us define the problem (Q) in the new variables $z \in R$, $x \in S$.

$$\begin{aligned} \min z \\ f_i(x) \leq z, \quad i = 1, \dots, m, \quad x \in S \end{aligned}$$

The feasible set of Problem (Q) can be defined as

$$D = \left\{ y = \begin{pmatrix} z \\ x \end{pmatrix} : f_i(x) \leq z, \quad i = 1, \dots, m, \quad x \in S \right\},$$

in correspondence to the objective function

$$g(y) = z.$$

Now, let us define the transformations

$$\rho(x) = \left(\max_{1 \leq i \leq m} \{f_i(x)\} \right)_x, \quad \sigma(y) = \sigma \left(\begin{pmatrix} z \\ x \end{pmatrix} \right) = x.$$

It can be easily verified that $\rho(x) \in D$ and that $\sigma(y) \in S$. Moreover, we have

$$g(\rho(x)) = \max_{1 \leq i \leq m} \{f_i(x)\} = f(x)$$

$$f(\sigma(y)) = f(x) = \max_{1 \leq i \leq m} \{f_i(x)\} \leq z = g(y),$$

so that the assumptions of the preceding proposition are satisfied.

Note that the equivalence considered above can be stated because of the fact that we *minimize* the function f . We cannot transform a maximization problem of the form

$$\max \left(\max_{1 \leq i \leq m} \{f_i(x)\} \right), \quad x \in S$$

into a problem with the same constraints of Problem (Q) and objective z to be maximized. In fact, in this case we would have feasible points with arbitrarily large values of z .

Some examples of equivalent transformations are given in Table 2.1.

Consider the first three problems; we can note that if we suppose that the functions f_i are linear that is

$$f_i(x) = c_i^T x \quad i = 1, \dots, m,$$

and that S is defined by linear constraints, all the transformed problems (Q) become linear programming problems, whereas the problems (P) are nonlinear non differentiable problems.

Table 2.1 Equivalent transformations

Problem (P)	Problem (Q)
$\min \max_{1 \leq i \leq m} \{f_i(x)\}$ $x \in S$	$\min z$ $f_i(x) \leq z, \quad i = 1, \dots, m$ $x \in S$
$\min \max_{1 \leq i \leq m} \{ f_i(x) \}$ $x \in S$	$\min z$ $-z \leq f_i(x) \leq z, \quad i = 1, \dots, m$ $x \in S$
$\min \sum_{i=1}^m f_i(x) $ $x \in S$	$\min \sum_{i=1}^m z_i$ $-z_i \leq f_i(x) \leq z_i, \quad i = 1, \dots, m$ $x \in S$
$\min f(x)$ $g(x) \geq b, \quad x \in S$	$\min f(x)$ $g(x) - w = b, \quad w \geq 0, \quad x \in S$
$\min f(x)$ $x \in S$	$\min f(x^+ - x^-)$ $x^+ \geq 0, \quad x^- \geq 0$ $x^+ - x^- \in S$
$\min f(x)$ $x \in S$	$\min z$ $f(x) \leq z, \quad x \in S$

2.5.3 Fractional Programming

Many optimization problems in engineering and in economics can be formulated as the minimization of a ratio between two functions, which measures the efficiency of a system, such as, for example, the ratio cost/time of a service. Problems of this form are termed *fractional programming problems*. See, e.g. [244]. We consider here a simple fractional programming problem where the objective function is the ratio of two affine functions and we show that, under appropriate assumptions, this problem can be transformed into an equivalent linear programming problem. Consider the nonlinear programming problem

$$\min \frac{c^T x + \beta}{a^T x + \delta}$$

$$Ax \geq b,$$

where A is an $m \times n$ real matrix, $c, a, x \in R^n$ and $b \in R^m$. Suppose that the following assumptions hold

- (h₁) $a^T x + \delta > 0$ for all $x \in R^n$ such that $Ax \geq b$;
- (h₂) there does not exist $y \in R^n$, $y \neq 0$ such that $Ay \geq 0$.

Letting

$$t = \frac{1}{a^T x + \delta},$$

we can consider the new problem in the variables t, x defined by

$$\begin{aligned} & \min t(c^T x + \beta) \\ & Ax \geq b, \\ & t(a^T x + \delta) = 1, \quad t > 0 \end{aligned}$$

which is still a nonlinear problem because of the product tx . Now we introduce the new vector $w = tx$ and we consider the linear programming problem in the variables $w \in R^n$ and $t \in R$ defined by:

$$\begin{aligned} & \min c^T w + \beta t \\ & Aw - bt \geq 0, \\ & a^T w + \delta t = 1, \quad t \geq 0. \end{aligned}$$

We can easily verify that under the assumptions (h₁) (h₂) the transformed problem is equivalent to the original problem. We note, in particular, that by (h₂) there do not exist feasible solutions of the transformed problem with $t = 0$. In fact, if $t = 0$ we have $Aw \geq 0$ and hence, by (h₂) we must have $w = 0$, which contradicts the constraint $a^T w + \delta t = 1$, so that every solution of the transformed problem must satisfy $t > 0$. Then, given an optimal solution of the transformed problem w^*, t^* we obtain the solution of the original problem by letting $x^* = w^*/t^*$.

2.5.4 Goal Programming

In many real applications in engineering and in economics we may have different conflicting objectives. As an example, we may require to minimize the cost of a transportation system and to minimize the service times or else to maximize the expected profit of a financial investment and to minimize the associated risks. Only in a few special cases we can determine an optimal solution with respect to all objectives. In general, we could define the concept of a *vector optimum* that

corresponds to the determination of the *non dominated solutions* or *Pareto efficient* solutions. See, e.g. [176].

More specifically, given an r -vector function $F(x) = (f_1(x), f_2(x), \dots, f_r(x))$, defined on a set $S \subseteq R^n$, we can say that $x^* \in S$ is a vector minimum point on S with respect to F , if there not exists $x \in S$ such that

$$F(x) \leq F(x^*), \quad \text{and } f_i(x) < f_i(x^*), \text{ for at least an } i.$$

However, even when non dominated solutions can be found, we typically have that there exist a very large number of these solutions and moreover, in many cases the distinction between “objectives” and “constraints” can be questionable. These difficulties have motivated the study of *multi-objective optimization* methods (see, eg. [233]) and there exists a quite large literature on theoretical aspects and heuristic computational methods of multi-objective optimization. Here we will confine ourselves to illustrate a simple example of a well known technique known as *goal programming* (see, e.g. [54]) that includes also different approaches.

Let us assume that r real functions $f_i : R^n \rightarrow R$ are given, defined on a set $S \subseteq R^n$. Without loss of generality we suppose that ideally we would minimize each function on S . As this could be impossible, we can attempt to define, for each function a desired value g_i^* that we call *goal* which, ideally, would be the maximum acceptable value for the objective f_i , that is, we would like to satisfy on S the conditions $f_i(x) \leq g_i^*, i = 1, \dots, r$. Imposing these conditions on S as *hard constraints* would typically determine an empty set (if the goals are too small) and hence we could attempt to minimize the violation of the goals, by solving the problem

$$\min_{x \in S} \sum_{i=1}^r \alpha_i \max\{0, f_i(x) - g_i^*\}$$

where $\alpha_i \geq 0$ is the weight assigned to the i -th goal. Recalling the equivalent transformation defined before, the problem can be equivalently rewritten in the form

$$\begin{aligned} \min & \sum_{i=1}^r \alpha_i z_i \\ & f_i(x) - g_i^* \leq z_i, \quad i = 1, \dots, r \\ & z_i \geq 0, \quad i = 1, \dots, r \\ & x \in S. \end{aligned}$$

An alternative formulation can be based on the introduction of *deviational variables* that measure both positive and negative violations of each goal. Assuming that we may have not only conditions expressed by imposing $f_i(x) \leq g_i^*$, but also

conditions of the form $f_i(x) \geq g_i^*$ or $f_i(x) = g_i^*$, we can write the goal programming problem in the form

$$\begin{aligned} \min & \sum_{i=1}^r \alpha_i d_i^+ + \beta_i d_i^- \\ & f_i(x) - g_i^* = d_i^+ - d_i^-, \quad i = 1, \dots, r \\ & d_i^+ \geq 0, \quad d_i^- \geq 0, \quad i = 1, \dots, r \\ & x \in S, \end{aligned}$$

where $\alpha_i, \beta_i \in R$ with $\alpha_i, \beta_i \geq 0$ are suitable weights. Then different goals can be expressed by an appropriate choice of the weights. More specifically, if we would force the condition $f_i(x) \leq g_i^*$ we can assume $\beta_i = 0$ and $\alpha_i > 0$; similarly, if we would require $f_i(x) \geq g_i^*$ we can set $\alpha_i = 0$ and $\beta_i > 0$ and finally if we would require $f_i(x) = g_i^*$ we can take $\alpha_i > 0, \beta_i > 0$.

If S is a polyhedral set, that is, if for some matrix $A(m \times n)$ and $b \in R^m$ we have $S = \{x \in R^n : Ax \geq b\}$ and if the objectives are linear

$$f_i(x) = c_i^T x, \quad c_i \in R^n, \quad i = 1, \dots, r,$$

then the goal programming problem in the variables $x, d_i^+, d_i^-, i = 1, \dots, r$ becomes a linear programming problem.

It is easily seen that the main difficulty can be that of choosing appropriately the weights, for scaling the different objectives and, possibly, for imposing suitable priorities between the goals. Thus interactive procedures are usually suggested in practical applications, for evaluating the effects of the choice of the weights.

2.6 Existence Conditions

Consider the nonlinear programming problem in R^n :

$$\begin{aligned} \min & f(x) \\ & x \in S, \end{aligned}$$

where $S \subseteq R^n$ and $f : S \rightarrow R$. One of the following situations may occur:

- the feasible set is empty: $S = \emptyset$;
- $S \neq \emptyset$, but f is unbounded from below on S , that is: $\inf_{x \in S} f(x) = -\infty$;
- $S \neq \emptyset$, $\inf_{x \in S} f(x) > -\infty$, but there not exist $x^* \in S$ such that $f(x^*) = \inf_{x \in S} f(x)$;
- there exists $x^* \in S$ such that $f(x^*) = \min_{x \in S} f(x)$.

We note, in particular, that “solving” our problem requires also establishing whether the feasible set is non empty, which can be, in general, a difficult problem of global nature, known as the *feasibility problem*.

An important result that gives in many cases a sufficient (but not necessary) existence condition is the following special case of Weierstrass Theorem , where we assume that f is continuous on S .

Proposition 2.7 (Weierstrass Theorem) *Let $f : R^n \rightarrow R$ be a continuous function and let $S \subset R^n$ be a non empty compact set. Then there exists a global minimum point of f in S .*

Proof Let $\ell = \inf_{x \in S} f(x)$, with $\ell \geq -\infty$, and let $\{x_k\}$ be a sequence of points $x_k \in S$ such that $\lim_{k \rightarrow \infty} f(x_k) = \ell$.

As S is compact there must exist a point $x^* \in S$ and a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K, k \rightarrow \infty} x_k = x^*, \quad \lim_{k \in K, k \rightarrow \infty} f(x_k) = \ell.$$

Then, the continuity of f implies $\lim_{k \in K, k \rightarrow \infty} f(x_k) = f(x^*)$, and hence we have $f(x^*) = \ell = \inf_{x \in S} f(x) = \min_{x \in S} f(x)$. \square

The preceding proposition can be directly applied to establish the existence of an optimal solution only in constrained problems, when the objective function is continuous and the feasible set is non empty and compact. More generally, however, we can make use of Weierstrass theorem with reference to suitable subsets of the feasible set.

In the sequel we illustrate first existence results in the unconstrained case and then we consider some important classes of constrained problems.

2.6.1 Existence in Unconstrained Optimization

In the unconstrained case the feasible set is the whole space R^n , which is obviously non empty but non compact. We introduce the following definitions.

Definition 2.5 (Level Set, Contour) Let $f : R^n \rightarrow R$ and $\alpha \in R$; we define (*sub*) *level set* or *inferior level set* of f on R^n a non empty set such that

$$\mathcal{L}(\alpha) = \{x \in R^n : f(x) \leq \alpha\}.$$

(continued)

Definition 2.5 (continued)

We define *contour* of f a non empty set such that

$$\mathcal{C}(\alpha) = \{x \in R^n : f(x) = \alpha\}.$$

Similarly, we can define *superior level set* a non empty set of the form

$$\{x \in R^n : f(x) \geq \alpha\}.$$

In the sequel, however the term “level set”, unless otherwise stated, will be referred to inferior level sets.

We can note that a (non empty) level set can be always constructed, by choosing arbitrarily $x_0 \in R^n$ and defining the set

$$\mathcal{L}^0 \equiv \mathcal{L}(x_0) = \{x \in R^n : f(x) \leq f(x_0)\}.$$

Some graphical representations of level sets and contours in one and two-dimensional cases are given below.

In Fig. 2.5 we consider a function defined on R . In this case, (in the region displayed in the figure) the level set corresponding to the indicated value of α is the union of the two intervals L_1, L_2 and the contour is the set of points where $f(x) = \alpha$.

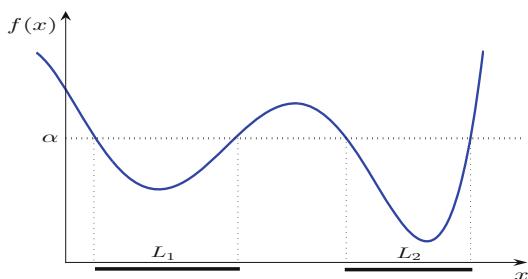
In Fig. 2.6 we show a three-dimensional representation of two functions defined in R^2 and for each function the contours are shown in a region of the x_1, x_2 plane.

In Fig. 2.7 we show the ellipsoidal contours of the strictly convex quadratic function

$$f(x) = x_1^2 + 10x_2^2.$$

The center $(0, 0)$ is obviously the unique global minimizer of f .

Fig. 2.5 A level set of a function $f : R \rightarrow R$



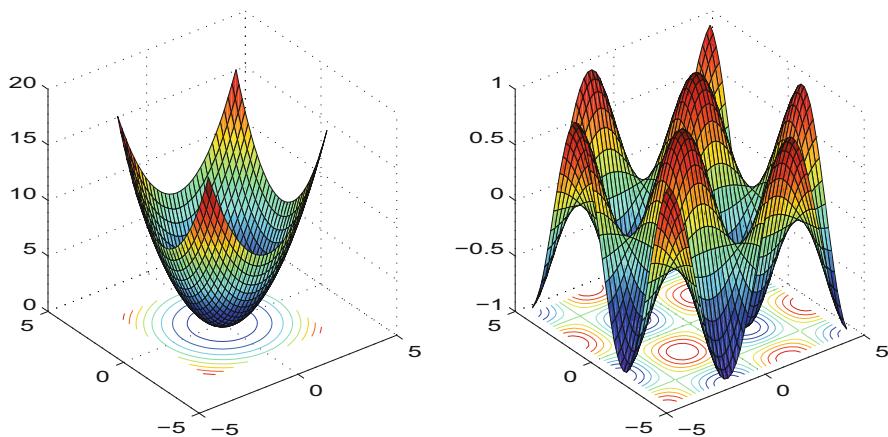


Fig. 2.6 Three-dimensional representations and contours of functions $f : \mathbb{R}^2 \rightarrow \mathbb{R}$

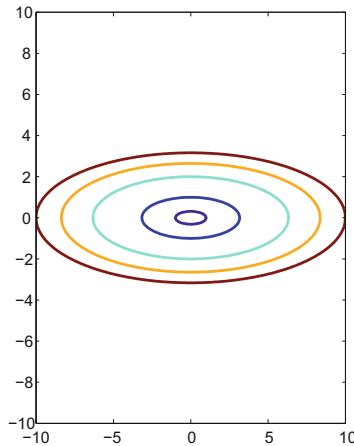


Fig. 2.7 Contours of a strictly convex quadratic function

A sufficient existence conditions for an unconstrained minimization problem can be established with reference to the level sets of the objective function. In fact, we can state the following condition.

Proposition 2.8 (Sufficient Existence Condition) *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuous function defined on \mathbb{R}^n and assume there exists a compact level set of f . Then there exists a global minimum point of f on \mathbb{R}^n .*

Proof Let $\mathcal{L}(\alpha)$ be a compact level set of f in R^n . Then, by definition, $\mathcal{L}(\alpha)$ is non empty and hence, from Proposition 2.7 it follows that there exists a point $x^* \in \mathcal{L}(\alpha)$, such that $f(x^*) \leq f(x) \leq \alpha$ for all $x \in \mathcal{L}(\alpha)$. On the other hand, we have that $x \notin \mathcal{L}(\alpha)$ implies that $f(x) > \alpha \geq f(x^*)$ and hence we can conclude that x^* is a global minimizer of f on R^n . \square

The next proposition yields a necessary and sufficient condition (known as *coercivity condition*) for the compactness of *all* level sets of f .

Proposition 2.9 (Compactness of All Level Sets, Coercivity) *Let $f : R^n \rightarrow R$ be a continuous function. Then all the level sets of f are compact if and only if f is coercive, that is, if and only if, for every sequence $\{x_k\}$ such that*

$$\lim_{k \rightarrow \infty} \|x_k\| = \infty,$$

we have $\lim_{k \rightarrow \infty} f(x_k) = \infty$.

Proof Necessity. Suppose first that all level sets of f are compact. Reasoning by contradiction, let us assume that there exist a sequence $\{x_k\}$ such that

$$\lim_{k \rightarrow \infty} \|x_k\| = \infty \tag{2.2}$$

and a number $\alpha \in R$ such that, for some subsequence (redefined $\{x_k\}$) we have $f(x_k) \leq \alpha$. This implies that $x_k \in \mathcal{L}(\alpha)$ for all k . But $\mathcal{L}(\alpha)$ is compact and hence bounded and this contradicts (2.2).

Sufficiency. Suppose now that f is coercive. As f is continuous, the level sets $\mathcal{L}(\alpha)$ are closed and we must show that they are also bounded. Suppose, by contradiction, that there exists a number $\hat{\alpha}$ such that the level set $\mathcal{L}(\hat{\alpha})$ is non empty and unbounded and let $\{x_k\}$ be a sequence of points in $\mathcal{L}(\hat{\alpha})$ such that $\|x_k\| \rightarrow \infty$. As, by assumption, f is coercive we must have

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

But this contradicts the assumption $x_k \in \mathcal{L}(\hat{\alpha})$, which implies $f(x_k) \leq \hat{\alpha}$ for all k . \square

We give here some examples of coercive functions.

Example 2.2 Consider the function $f : R^n \rightarrow R$ defined by:

$$f(x) = \|x\|_a^s,$$

where $\|\cdot\|_a$ is any norm and $s > 0$. As all norms on R^n are equivalent the assertion follows directly from the definition. \square

Example 2.3 A function of the form $f(x) = E(x) + \tau \|x\|^2$, where $E(x) \geq 0$ for all x and $\tau > 0$, is obviously coercive. \square

Example 2.4 Let $f : R^n \rightarrow R$ be a quadratic function of the form

$$f(x) = \frac{1}{2}x^T Qx + c^T x,$$

where Q is a symmetric positive definite matrix. Let $\{x_k\}$ be a sequence in R^n . Denoting by $\lambda_m(Q)$ the smallest eigenvalue of Q , we can write

$$f(x_k) \geq \frac{\lambda_m(Q)}{2}\|x_k\|^2 - \|c\|\|x_k\| = \left(\frac{\lambda_m(Q)}{2}\|x_k\| - \|c\|\right)\|x_k\|.$$

If $\{x_k\}$ satisfies $\lim_{k \rightarrow \infty} \|x_k\| = \infty$, as $\lambda_m(Q) > 0$, we get

$$\frac{\lambda_m(Q)}{2}\|x_k\| - \|c\| > 0,$$

for sufficiently large values of k and hence we have $\lim_{k \rightarrow \infty} f(x_k) = \infty$, which shows that f is coercive on R^n . \square

Example 2.5 Consider the function $f : R^n \rightarrow R$ defined by

$$f(x) = \|Ax - b\|_a^s$$

where A is an $m \times n$ matrix with rank n , $b \in R^m$, $\|\cdot\|_a$ is any norm on R^n and $s > 0$. It can be easily shown that the function f is coercive.

Consider first the case of the Euclidean norm $\|\cdot\|$. Then, define the quadratic function

$$q(x) = 1/2\|Ax - b\|^2,$$

with Hessian matrix $A^T A$. As the matrix A has rank n , the Hessian matrix is symmetric positive definite and hence, recalling the preceding example, we know that the function q is coercive.

On the other hand, because of the equivalence of the norms on R^n there must exist $c > 0$ such that, for all $x \in R^n$ we have

$$(2q(x))^{1/2} = \|Ax - b\| \leq c\|Ax - b\|_a,$$

and hence, if A has rank n the function f is coercive in any norm. \square

Example 2.6 Suppose now that f is a twice continuously differentiable function uniformly convex on R^n , that is, such that the Hessian matrix of f satisfies the condition

$$d^T \nabla^2 f(x) d \geq m \|d\|^2 \quad \text{for all } x, d \in R^n,$$

with $m > 0$. Then f is coercive on R^n . In fact, if $x_0 \in R^n$ and x is a point in R^n , by Taylor's theorem we can write

$$f(x) = f(x_0) + \nabla f(x_0)^T (x - x_0) + \frac{1}{2} (x - x_0)^T \nabla^2 f(z) (x - x_0),$$

where

$$z = x_0 + \xi(x - x_0) \quad \text{with} \quad \xi \in (0, 1).$$

By Cauchy-Schwarz inequality and by the uniform convexity assumption, we have

$$f(x) \geq f(x_0) - \|\nabla f(x_0)\|_2 \|x - x_0\| + \frac{m}{2} \|x - x_0\|^2,$$

which implies the coercivity of f . \square

By employing the coercivity property, from Propositions 2.8 and 2.9 we obtain immediately the following sufficient existence condition.

Proposition 2.10 (Sufficient Existence Condition) *Let $f : R^n \rightarrow R$ be a continuous and coercive function on R^n . Then there exists a global minimum point of f on R^n .* \square

As already remarked, the coercivity condition is a sufficient, but not necessary condition for the existence of a global minimum point.

An important example is the *linear least squares problem*

$$\min f(x) \equiv \|Ax - b\|^2, \quad x \in R^n,$$

where A is a $m \times n$ matrix and $b \in R^m$. If $\text{rank}(A) < n$ then the function f is not coercive, but it can be shown that the linear least squares problem always admits a global solution.

Actually we can establish a more general result, given in the next proposition, which can be proved by employing different techniques; here we give a simple direct proof.

Proposition 2.11 Let A a $m \times n$ matrix and let $b \in R^m$. The problem

$$\min f(x) \equiv \|Ax - b\|_a^q, \quad x \in R^n$$

where $\|\cdot\|_a$ is any norm on R^n and $q > 0$, admits a global optimal solution in R^n .

Proof We suppose that $\text{rank}(A) < n$, for, otherwise, the result follows from the preceding examples.

Then, after a reordering, if needed, we can partition the matrix in the form

$$A = (B \quad C)$$

where B is a sub-matrix $m \times n_1$ of rank $n_1 < n$, and the columns of C of dimension $m \times n_2$, can be expressed as linear combinations of the columns of B . If C_i is the i -th column of C , we can write

$$C_i = B\Gamma_i,$$

where Γ_i is a vector in R^{n_1} . Thus, there exists a matrix Γ ($n_1 \times n_2$) such that

$$C = B\Gamma.$$

Given a vector $x \in R^n$, we can partition it into the components $x(1) \in R^{n_1}, x(2) \in R^{n_2}$ and hence we can write

$$\|Ax - b\|_a^q = \|Bx(1) + Cx(2) - b\|_a^q = \|B(x(1) + \Gamma x(2)) - b\|_a^q.$$

Letting

$$y = x(1) + \Gamma x(2)$$

we have

$$\|Ax - b\|_a^q = \|By - b\|_a^q.$$

Consider now the problem

$$\min \|By - b\|_a^q, \quad y \in R^{n_1}. \quad (2.3)$$

As B has rank n_1 , the function $\|By - b\|_a^q$ is coercive on R^{n_1} and hence it admits a global minimum point $y^* \in R^{n_1}$, such that

$$\|By^* - b\|_a^q \leq \|By - b\|_a^q, \quad \text{for all } y \in R^{n_1}. \quad (2.4)$$

We can define the point $x^* \in R^n$ such that $x^*(1) = y^*$, $x^*(2) = 0$, so that we obtain

$$\|Ax^* - b\|_a^q = \|By^* - b\|_a^q.$$

We claim that x^* is a global minimizer of $\|Ax - b\|_a^q$.

Reasoning by contradiction, suppose there exists $\hat{x} \in R^n$, with vector components $\hat{x}(1), \hat{x}(2)$ such that

$$\|A\hat{x} - b\|_a^q < \|Ax^* - b\|_a^q.$$

Then, we can consider the point

$$\hat{y} = \hat{x}(1) + \Gamma \hat{x}(2),$$

so that we can write:

$$\|A\hat{x} - b\|_a^q = \|B\hat{y} - b\|_a^q.$$

It follows that we must have

$$\|B\hat{y} - b\|_a^q < \|Ax^* - b\|_a^q = \|By^* - b\|_a^q,$$

which contradicts (2.4). This concludes the proof. \square

2.6.2 Existence in Constrained Optimization

We consider constrained problem of the form

$$\min f(x) \quad x \in S,$$

where $S \subset R^n$ and $f : S \rightarrow R$ is assumed to be at least continuous.

A first possibility for establishing the existence of an optimal solution is obviously that of showing that the feasible set S is non empty and compact. In some cases this can be directly recognized by inspection.

Examples are problems with a simple structure of the constraints, such as

- problems *with box constraints*:

$$S = \{x \in R^n : a_i \leq x_i \leq b_i, \quad i = 1, \dots, n\},$$

where $a_i \leq b_i, \quad i = 1, \dots, n$;

- problems with *simplex constraints*:

$$S = \{x \in R^n : e^T x \leq 1, x \geq 0\},$$

where $e^T = (1, \dots, 1)$ is the unit vector in R^n ;

- problems defined on a spherical region in any norm, such as:

$$S = \{x \in R^n : \|x\|_a \leq r\},$$

for some $r > 0$.

Existence conditions, based on the level sets considered in the unconstrained case, can be extended to the constrained case. In particular, if S is non empty, a sufficient existence condition is that of requiring that there exists $\alpha \in R$ such that the corresponding *level set on S* , that is a *non empty* set defined by

$$\mathcal{L}_S(\alpha) = S \cap \{x \in R^n : f(x) \leq \alpha\} \equiv \{x \in S : f(x) \leq \alpha\}$$

is compact.

Note that, if S is non empty, a level set can be constructed, in principle, by choosing a feasible point $x_0 \in S$ and then defining the set

$$\mathcal{L}_S^0 \equiv \mathcal{L}_S(f(x_0)) = \{x \in S : f(x) \leq f(x_0)\}.$$

In order to extend the results given in the unconstrained case, we suppose that S is non empty, the function f is continuous on S and the set $\mathcal{L}_S(\alpha)$ is a (non empty) level set. First we give the definition of *coercive function on a set*.

Definition 2.6 A function $f : S \rightarrow R$ is said to be *coercive on $S \subset R^n$* if, for every sequence $\{x_k\}$ such that $x_k \in S$ and $\lim_{k \rightarrow \infty} \|x_k\| = \infty$, we have

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

□

Note that, on the basis of our definition, if S is a bounded set then, any function defined on S is *vacuously* coercive.

In the next proposition we do not assume that S is closed. Actually, in some important classes of problems the feasible set S is not closed and the objective function contains barrier terms that go to infinity on the boundary of S . In this case we can still guarantee that the level sets on S are compact (and hence that there exists a global minimizer) by imposing suitable conditions on the behaviour of f on the boundary of S .

Proposition 2.12 (Compactness of All Level Sets on S) Suppose that $S \subseteq R^n$ is a non empty set and that $f : S \rightarrow R$ is a continuous function on S . Then all the level sets of f on S are compact if and only if the following conditions hold

- (i) f is coercive on S , that is for every sequence $\{x_k\}$ such that $x_k \in S$ and $\lim_{k \rightarrow \infty} \|x_k\| = \infty$, we have

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

- (ii) for every sequence $\{x_k\}$ such that $x_k \in S$ and $\lim_{k \rightarrow \infty} x_k = \bar{x} \notin S$, we have

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

Proof Necessity. We suppose first that all level sets of f on S are compact and we show, reasoning by contradiction, that conditions (i) and (ii) must hold. Assume first that (i) is not true and hence that there exist a sequence $\{x_k\}$ of points $x_k \in S$, such that

$$\lim_{k \rightarrow \infty} \|x_k\| = \infty, \tag{2.5}$$

and a number $\bar{\alpha} \in R$ such that, for some subsequence (redefined $\{x_k\}$) we have $f(x_k) \leq \bar{\alpha}$. This implies that $x_k \in \mathcal{L}_S(\bar{\alpha})$ for all k . But, by assumption, $\mathcal{L}_S(\bar{\alpha})$ is compact and hence bounded, and this contradicts (2.5).

Similarly, if we suppose that (ii) is false, we can repeat the same reasonings and conclude that we contradict the assumption that the level sets on S are closed.

Sufficiency. Suppose now that conditions (i) and (ii) hold. First we must show that the level sets $\mathcal{L}_S(\alpha)$ are closed. Suppose the contrary, and hence that there exists a level set $\mathcal{L}_S(\bar{\alpha})$ which is not closed, so that for some sequence $\{x_k\}$ such that $x_k \in \mathcal{L}_S(\bar{\alpha})$, we have $\lim_{k \rightarrow \infty} x_k = \bar{x} \notin \mathcal{L}_S(\bar{\alpha})$. As $f(x_k) \leq \bar{\alpha}$ and f is continuous

we must have necessarily that $\bar{x} \notin S$. Then by (ii) we get a contradiction with the assumption that $f(x_k) \leq \bar{\alpha}$ for all k .

Now we must show that the level sets $\mathcal{L}_S(\alpha)$ are bounded. Reasoning again by contradiction, suppose there exists a number $\hat{\alpha}$ such that the level set $\mathcal{L}_S(\hat{\alpha})$ is non empty and unbounded and let $\{x_k\}$ be a sequence of points in $\mathcal{L}_S(\hat{\alpha})$ such that $\|x_k\| \rightarrow \infty$. By condition (i) we must have

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

But this contradicts the assumption $x_k \in \mathcal{L}_S(\hat{\alpha})$, which implies $f(x_k) \leq \hat{\alpha}$ for all $x_k \in S$. \square

Now, from the preceding results we easily obtain the following sufficient existence condition, which extends the result of Proposition 2.10.

Proposition 2.13 (Sufficient Existence Condition) *Let $S \subseteq R^n$ be a non empty set and let $f : S \rightarrow R$ be a function continuous on S . Suppose that conditions (i) and (ii) of Proposition 2.12 are satisfied. Then there exists a global minimum point of f on S .* \square

Note that if S is closed, condition (ii) is vacuously satisfied and hence the sufficient condition (for S non empty) is that f is coercive on S .

Example 2.7 Let $z \in R^n$ a given point, let $S \subseteq R^n$ a non empty closed set and consider the problem of determining a point of S at minimal distance from z , that is the problem

$$\begin{aligned} \min & \|x - z\| \\ x & \in S. \end{aligned}$$

The function $f(x) = \|x - z\|$ is obviously coercive on S and hence by Proposition 2.13 there exists a global minimum point of f on S . \square

2.6.3 Feasibility Problems

In the existence conditions considered before we have always assumed that the feasible set is non-empty. Recognizing this may be quite difficult, in practice, since in the general case, establishing feasibility can be equivalent to solving a global optimization problem, even if we would be happy with determining only a critical point of the constrained problem.

As we know, the feasible set in nonlinear optimization is typically described by a system of equations and inequalities:

$$S = \{x \in R^n : h(x) = 0, \quad g(x) \leq 0\},$$

where $h : R^n \rightarrow R^p$, and $g : R^n \rightarrow R^m$. Thus, establishing feasibility requires establishing whether a system of (possibly nonlinear) equations and inequalities has a solution, which is a difficult problem in Nonlinear Analysis.

We must distinguish two classes of methods for solving a nonlinear programming problem:

- (a) methods that require the knowledge of a starting feasible point $x_0 \in S$;
- (b) *exterior* methods that should reach the feasible set only in the limit of a sequence.

In case (a) we may attempt to compute a feasible point by solving a *phase one problem*.

One possibility, as we know, is that of defining a merit function and then computing, when possible, a global minimizer of this function. As an example, we could define the merit function

$$\Psi(x) = \sum_{i=1}^p (h_i(x))^2 + \sum_{i=1}^m (\max \{0, g_i(x)\})^2.$$

If Ψ has an optimal solution and we can find a global minimizer x_0 , it is easily seen that if $\Psi(x_0) = 0$ the point x_0 is a feasible point of the constrained problem, while if $\Psi(x_0) > 0$ we can conclude that the given problem is infeasible. However, if we minimize Ψ using the algorithms that search for a critical point of Ψ we must assume that suitable (quite strong) conditions on the constraints and on the merit function are satisfied, in order to guarantee that a feasible point exists and can be found. Some examples will be given in the analysis of constrained algorithms.

An alternative approach can be that of defining an auxiliary constrained problem where we introduce suitable *artificial variables* and an *artificial objective function*. In particular, if the feasible set is defined by *linear constraints* (which we assume in *standard form*), that is

$$S = \{Ax = b, x \geq 0\}, \quad \text{with } b \geq 0,$$

an artificial problem, which yields (if it exists) a feasible point, would be the linear programming problem,

$$\begin{aligned} \min \quad & \sum_{j=1}^n y_j \\ \text{subject to} \quad & Ax + y = b \\ & x \geq 0, y \geq 0, \end{aligned}$$

where we have introduced a vector $y \in R^n$ of artificial variables. In this case $x = 0, y = b$ is a feasible point of the artificial problem, which can be solved through a linear programming algorithm. Then we have obviously that our constrained problem would admit a feasible solution if and only if the artificial problem has an optimal solution \bar{x}, \bar{y} with $\bar{y} = 0$.

In case (b) we do not need a starting feasible point and we take feasibility into account through suitable penalty terms in the objective function.

Methods based in this approach will be considered in the study of penalty methods in Chap. 21. Also in this case, however, suitable assumptions on the constraints must be imposed in order to guarantee that S is non empty.

2.7 Exercises

2.1 Consider the problem:

$$\begin{array}{ll} \min & f(x) \\ x \in S, & \end{array}$$

and assume that there exists a global minimum point. Show that the problem

$$\begin{array}{ll} \min & cf(x) + d \\ x \in S, & \end{array}$$

with $c > 0$, admits the same (local and global) minimizers of the given problem.

2.2 Consider the problem:

$$\begin{array}{ll} \min & f(x) \\ x \in S, & \end{array}$$

where $f : S \rightarrow D \subseteq R$ and assume that there exists a global minimum point. Show that the problem

$$\begin{array}{ll} \min & F[f(x)] \\ x \in S, & \end{array}$$

where $F : D \rightarrow R$ is a monotone increasing function, admits the same (local and global) minimizers of the given problem.

2.3 Let A be a $m \times n$ matrix and $b \in R^m$. Prove that the function

$$f(x) = \|Ax - b\|,$$

where $\|\cdot\|$ is any norm, is coercive if and only if the columns of A are linearly independent.

2.4 Prove the Weierstrass Theorem by assuming that function f is lower semicontinuous.

(A function is lower semicontinuous at \bar{x} if

$$\liminf_{k \rightarrow \infty} f(x_k) \geq f(\bar{x})$$

for any sequence $\{x_k\}$ convergent to \bar{x} .

2.5 Let $f : R^n \rightarrow R$. Show that a point x^* is a global minimum point of f if and only if for every $d \in R^n$ the function $g : R \rightarrow R$, defined by $g(\alpha) = f(x^* + \alpha d)$, is such that the point $\alpha^* = 0$ is a global minimum point.

2.6 Consider the linear programming problem:

$$\min c^T x$$

$$Ax \leq b$$

$$x \geq 0,$$

define an equivalent linear programming problem with equality constraints and non-negativity constraints, and prove the equivalence.

2.7 Consider the problem:

$$\min f(x)$$

$$x \geq 0, \quad x \in R^n.$$

Set

$$x_i = y_i^2, \quad i = 1, \dots, n$$

in the objective function and prove that the problem:

$$\min f(x(y))$$

$$y \in R^n,$$

where $x(y)$ is the point defined by the preceding transformation , is equivalent to the given problem.

2.8 Given the problem with *box* constraints:

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & a \leq x \leq b, \quad x \in R^n, \end{aligned}$$

where $a, b \in R^n$, consider the transformation:

$$x_i = \frac{1}{2}(b_i + a_i) + \frac{1}{2}(b_i - a_i) \sin(y_i), \quad i = 1, \dots, n;$$

show the equivalence between the (unconstrained) transformed problem and the original problem.

2.9 Consider the problem:

$$\begin{aligned} \min & |f(x)| \\ \text{s.t.} & x \in S \subseteq R^n \end{aligned}$$

and show its equivalence with the problem (in the z, x variables):

$$\begin{aligned} \min & z \\ \text{s.t.} & x \in S \\ & -z \leq f(x) \leq z. \end{aligned}$$

2.8 Notes and References

The definitions and the results introduced in this chapter are basic material in every introductory book. We mention here some of the reference texts that are of special relevance to this book, in the field of nonlinear finite dimensional continuous optimization, in addition to the books mentioned in the introduction: [12, 15, 18, 31, 67, 92, 93, 109, 206, 231] and [50, 51, 114, 152, 154, 162, 177, 183, 208, 221, 246, 260].

Chapter 3

Optimality Conditions for Unconstrained Problems in R^n



In this chapter we establish the basic *necessary conditions* and *sufficient conditions* for a point $x^* \in R^n$ to be an optimal local solution of the unconstrained problem

$$\text{minimize } f(x), \quad x \in R^n,$$

under the assumption that f is one or two times continuously differentiable.

We start by defining the concept of *descent direction* at a given point and then we deduce *first* and *second order* necessary optimality conditions by imposing that there not exist descent directions at a local minimizer. Then we show that points satisfying necessary conditions, which will be called *critical points*, can be proved to be also (local or global) minimizers under appropriate convexity assumptions.

As we already know, the optimality conditions play a major role not only in the theoretical study of optimization problems, but also in the definition and in the analysis of optimization algorithms, which are essentially constructed for approximating critical points. In particular, all the unconstrained methods described in the sequel are based on the necessary optimality conditions reported in this chapter.

3.1 Descent Directions

We introduce the following definition.

Definition 3.1 (Descent Direction) Let $f : R^n \rightarrow R$, $x \in R^n$ and $d \in R^n$, $d \neq 0$. We say that d is a descent direction for f at x if there exists a number $\tilde{t} > 0$ such that

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

□

A descent direction at x is then a nonzero direction such that every sufficiently small step from x along d corresponds to a strict decrease of the objective function. Similarly, we can define d as an *ascent direction* at x if f strictly increases in correspondence to sufficiently small steps along d .

Under the assumption that f is continuously differentiable, we can give a useful characterization of descent directions based on first order derivatives.

Proposition 3.1 (First Order Descent Condition) Suppose that $f : R^n \rightarrow R$ is continuously differentiable in a neighborhood of $x \in R^n$ and let $d \in R^n$ be a non zero vector. Then, if

$$\nabla f(x)^T d < 0, \tag{3.1}$$

the direction d is a descent direction for f at x .

Conversely, if f is continuously differentiable and convex in an open ball $B(x; \rho)$, with $\rho > 0$, and if d is a descent direction at x , then condition (3.1) must necessarily hold.

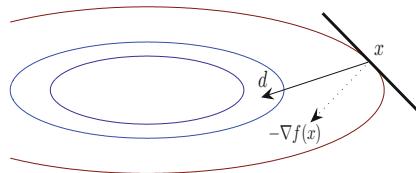
Proof Under the assumptions stated, the directional derivative of f at x is given by

$$\lim_{t \rightarrow 0^+} \frac{f(x + td) - f(x)}{t} = \nabla f(x)^T d. \tag{3.2}$$

Therefore, if $\nabla f(x)^T d < 0$ Eq. (3.2) implies that, for sufficiently small values of $t > 0$, we must have, by continuity: $f(x + td) - f(x) < 0$, so that d is a descent direction.

If f is convex on the convex set $B(x; \rho)$, for all sufficiently small values of $t \in (0, 1)$ we have that $x + td \in B(x; \rho)$ and that $f(x + td) \geq f(x) + t \nabla f(x)^T d$. Hence $\nabla f(x)^T d \geq 0$ would imply that $f(x + td) \geq f(x)$ and thus, if d is a descent direction, we must necessarily have $\nabla f(x)^T d < 0$. □

Fig. 3.1 Descent direction d at x



The above proposition implies that the condition $\nabla f(x)^T d < 0$ is a *sufficient condition* for ensuring that d is a descent direction. Similarly, if we have $\nabla f(x)^T d > 0$ we can say that d is an ascent direction. If $\nabla f(x)^T d = 0$ and hence, in particular, if $\nabla f(x) = 0$, we cannot decide about the nature of d in the absence of further information.

From a geometric point of view, recalling that the *angle* θ between $d \neq 0$ and $\nabla f(x) \neq 0$ can be defined by letting

$$\cos \theta = \frac{\nabla f(x)^T d}{\|\nabla f(x)\| \|d\|},$$

we can say that the angle between d and $\nabla f(x)$ is *obtuse* if $\nabla f(x)^T d < 0$ and it is *acute* if $\nabla f(x)^T d > 0$. If $\nabla f(x)^T d = 0$ the vectors d and $\nabla f(x)$ are *orthogonal*.

A descent direction is shown in Fig. 3.1, with reference to the contours of the lower level sets of a function in R^2 . We note that the direction opposite to the gradient vector, that is the direction $d = -\nabla f(x)$, is always a descent direction if $\nabla f(x) \neq 0$. In fact, in this case we have:

$$\nabla f(x)^T d = -\nabla f(x)^T \nabla f(x) = -\|\nabla f(x)\|^2 < 0.$$

When f is non convex we may have descent directions d at x such that $\nabla f(x)^T d = 0$. In particular, if x is a strict local maximizer, every non zero direction is a descent direction at x .

A better characterization of the descent properties can be obtained when f is two times continuously differentiable. First we give the following definition.

Definition 3.2 (Direction of Negative Curvature) Let $f : R^n \rightarrow R$ be two times continuously differentiable in an open neighborhood of $x \in R^n$. We say that $d \in R^n$, $d \neq 0$ is a direction of negative curvature for f at x if $d^T \nabla^2 f(x) d < 0$. \square

A direction of negative curvature is then a direction such that the second order directional derivative is negative, which implies that the first order directional derivative is strictly decreasing along d in a neighborhood of x .

In the next proposition we give a second order characterization of descent directions, based on directions of negative curvature.

Proposition 3.2 (Second Order Descent Condition) *Let $f : R^n \rightarrow R$ be two times continuously differentiable in an open neighborhood of $x \in R^n$ and let $d \in R^n$ be a non zero vector. Suppose that $\nabla f(x)^T d = 0$, and that d is a direction of negative curvature at x , that is $d^T \nabla^2 f(x)d < 0$. Then d is a descent direction for f at x .*

Proof As f is two times continuously differentiable we can write:

$$f(x + td) = f(x) + t \nabla f(x)^T d + \frac{1}{2}t^2 d^T \nabla^2 f(x)d + \beta(x, td)$$

where

$$\lim_{t \rightarrow 0} \frac{\beta(x, td)}{t^2} = 0.$$

Since, by assumption, we have $\nabla f(x)^T d = 0$, we can write

$$\frac{f(x + td) - f(x)}{t^2} = \frac{1}{2}d^T \nabla^2 f(x)d + \frac{\beta(x, td)}{t^2}$$

and hence, as $d^T \nabla^2 f(x)d < 0$ and $\beta(x, td)/t^2 \rightarrow 0$ for $t \rightarrow 0$, for sufficiently small values of t we have $f(x + td) - f(x) < 0$, and therefore d is a descent direction. \square

3.2 Optimality Conditions

In the next proposition we state an obvious consequence of the definition of descent direction.

Proposition 3.3 *Let $x^* \in R^n$ be a local minimum point of f in R^n . Then there cannot exist descent directions for f at x .*

Proof If d were a descent direction at x^* , then, by definition, we could find in every neighborhood of x^* a point $x^* + td$ such that $f(x^* + td) < f(x^*)$, and this would contradict the assumption that x^* is a local minimizer. \square

Under differentiability assumptions, recalling the first order descent condition of the preceding section, we get immediately the following proposition.

Proposition 3.4 (First Order Necessary Optimality Condition) *Let $f : R^n \rightarrow R$ be continuously differentiable in an open neighborhood of $x^* \in R^n$.*

Then, if x^ is an unconstrained local minimum point of f , the point x^* is a stationary point of f , that is:*

$$\nabla f(x^*) = 0.$$

Proof If $\nabla f(x^*) \neq 0$ the direction $d = -\nabla f(x^*)$ satisfies

$$d^T \nabla f(x^*) = -\|\nabla f(x^*)\|^2 < 0.$$

Therefore d is a descent direction and this yields a contradiction, as shown in the preceding proposition, \square

If f is two times continuously differentiable we can improve the optimality conditions, making use of second derivatives.

Proposition 3.5 (Second Order Necessary Optimality Condition) *Let $f : R^n \rightarrow R$ be two times continuously differentiable in an open neighborhood of $x^* \in R^n$.*

Then, if x^ is an unconstrained local minimum point of f , the point x^* is a stationary point of f and the Hessian matrix of f is positive semidefinite at x^* , that is:*

- (a) $\nabla f(x^*) = 0$;
- (b) $y^T \nabla^2 f(x^*)y \geq 0$, for all $y \in R^n$.

Proof Assertion (a) follows from Proposition 3.4. If (b) does not hold there must exist $y \in R^n$ such that $y^T \nabla^2 f(x^*)y < 0$, and hence, as $\nabla f(x^*)^T y = 0$, by Proposition 3.2, the direction y is a descent direction for f at x^* , which contradicts the assumption that x^* is a local minimum point. \square

We note that (a) and (b) of the preceding proposition are not sufficient conditions for proving that x^* is a local minimizer. Sufficient conditions can be obtained by imposing additional local convexity assumptions on f . In particular, we can state the following proposition.

Proposition 3.6 (Second Order Sufficient Optimality Condition) *Let $f : R^n \rightarrow R$ be two times continuously differentiable in the open convex neighborhood $B(x^*; \rho)$ of $x^* \in R^n$. Suppose that:*

- (i) $\nabla f(x^*) = 0$;
- (ii) $\nabla^2 f(x)$ is positive semidefinite on $B(x^*; \rho)$, that is

$$y^T \nabla^2 f(x)y \geq 0 \quad \text{for all } x \in B(x^*; \rho) \text{ and all } y \in R^n.$$

Then x^ is an unconstrained local minimum point of f .*

Proof Using Taylor's Theorem, as $\nabla f(x^*) = 0$, we can write for every $x \in B(x^*; \rho)$

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(x^* + \xi(x - x^*))(x - x^*),$$

where $\xi \in (0, 1)$. As $B(x^*; \rho)$ is convex, we have $x^* + \xi(x - x^*) \in B(x^*; \rho)$ and hence, by assumption (ii), the matrix $\nabla^2 f(x^* + \xi(x - x^*))$ is positive semidefinite. It follows that we have $(x - x^*)^T \nabla^2 f(x^* + \xi(x - x^*))(x - x^*) \geq 0$, and hence we have $f(x) \geq f(x^*)$, which establishes our thesis. \square

An immediate consequence is that if $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is *positive definite* then x^* is a local minimum point. In fact, in this case the assumptions of the preceding proposition are satisfied by continuity. However, when $\nabla^2 f(x^*)$ is positive definite, we can give a stronger result.

Proposition 3.7 (Sufficient Condition for a Strict Minimizer) *Let $f : R^n \rightarrow R$ be two times continuously differentiable in an open neighborhood of $x^* \in R^n$. Suppose that:*

- (i) $\nabla f(x^*) = 0$;
- (ii) $\nabla^2 f(x^*)$ is positive definite.

Then x^ is a strict local unconstrained minimum point of f ; moreover there exist $\rho > 0$ and $\mu > 0$ such that*

$$f(x^*) \leq f(x) - \frac{\mu}{2} \|x - x^*\|^2 \quad \text{for all } \|x - x^*\| < \rho.$$

Proof By continuity there exists a neighborhood $B(x^*; \rho_1)$ where $\nabla^2 f(x)$ is continuous and positive definite and the smallest eigenvalue $\lambda_{\min}(\nabla^2 f(x))$ of

$\nabla^2 f(x)$) is positive. Therefore, by choosing $\rho < \rho_1$ and taking into account the continuity of the eigenvalues with respect to x , there must exist $\mu > 0$ such that:

$$\mu = \min_{\|x-x^*\| \leq \rho} \lambda_{\min}(\nabla^2 f(x)) > 0.$$

Assuming that $\|x-x^*\| < \rho$, using Taylor's theorem and recalling that $\nabla f(x^*) = 0$, we can write

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(w)(x - x^*),$$

where $w = x^* + \xi(x - x^*)$ with $\xi \in (0, 1)$. As $B(x^*; \rho)$ is a convex set, we have $w \in B(x^*; \rho)$ and then we obtain

$$f(x) \geq f(x^*) + \frac{1}{2}\lambda_{\min}(\nabla^2 f(w))\|x - x^*\|^2 \geq f(x^*) + \frac{\mu}{2}\|x - x^*\|^2,$$

which proves the assertion. \square

From the preceding conditions we can easily derive necessary conditions and sufficient conditions for a point x^* to be a local maximizer. In particular, we can say that

- a first order necessary condition for x^* to be a unconstrained local maximizer is again that $\nabla f(x^*) = 0$;
- a second order necessary condition for x^* to be an unconstrained local maximizer is that $\nabla f(x^*) = 0$ and that $\nabla^2 f(x^*)$ is *negative semidefinite*;
- a sufficient condition for x^* to be an unconstrained local maximizer is that $\nabla f(x^*) = 0$ and that $\nabla^2 f(x)$ is *negative semidefinite* in a neighborhood of x^* ;
- a sufficient condition for x^* to be an unconstrained strict local maximizer is that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is *negative definite*;
- if $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is *(negative or positive) semidefinite* we cannot determine the nature of x^* in the absence of further information.

Example 3.1 Consider the function $f : R \rightarrow R$ defined by $f(x) = x^3$, which is infinitely differentiable on R . At the point $x^* = 0$ we have

$$\frac{df(x^*)}{dx} = 0, \quad \frac{d^2 f(x^*)}{d^2 x} = 0,$$

and hence both first order and second order necessary optimality conditions are satisfied. However, the point $x^* = 0$ is not a local minimizer, but it is an *inflection point* of f . \square

Example 3.2 Consider the function $f : R \rightarrow R$ defined by

$$f(x) = |x|^3.$$

The function is two times continuously differentiable and we have

$$\frac{df(x)}{dx} = \begin{cases} 3x^2 & x \geq 0 \\ -3x^2 & x \leq 0 \end{cases} \quad \frac{d^2 f(x)}{dx^2} = \begin{cases} 6x & x \geq 0 \\ -6x & x \leq 0 \end{cases}$$

The point $x^* = 0$ satisfies the necessary optimality conditions

$$\frac{df(x^*)}{dx} = 0, \quad \frac{d^2 f(x^*)}{dx^2} = 0.$$

As $f(x) > f(x^*) = 0$ for every $x \neq 0$, we can conclude that x^* is the unique global minimum point of f . We can note that f is a strictly convex function, but the second order derivative is not positive at x^* . \square

When $\nabla f(x^*) = 0$ and the Hessian matrix $\nabla^2 f(x^*)$ is *indefinite* (that is, there exists vectors, $u, v \in R^n$ such that $u^T \nabla^2 f(x^*) u > 0$ and $v^T \nabla^2 f(x^*) v < 0$) we can exclude that x^* is a local maximum or minimum point and x^* is a *saddle point*. At this point we can find both descent and ascent directions.

Example 3.3 Consider the function $f : R^2 \rightarrow R$ defined by

$$f(x) = x_1^2 - x_2^2.$$

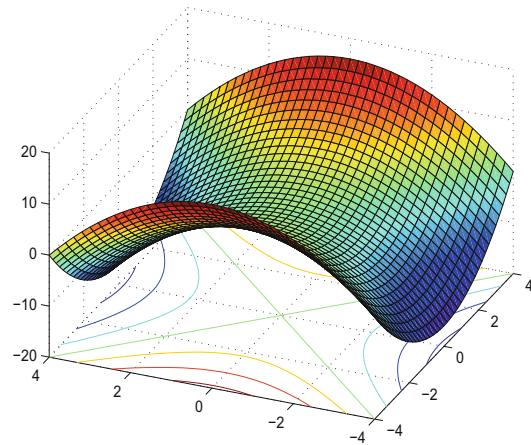
Letting $x^* = 0$, we have

$$\nabla f(x^*) = \begin{pmatrix} 2x_1^* \\ 2x_2^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \nabla^2 f(x^*) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}.$$

As $\nabla^2 f(x^*)$ is indefinite, f has a saddle point at the origin. We show in Fig. 3.2 a three-dimensional representation of f near the origin. We can note that f increases for increasing values of $|x_1|$ and decreases for increasing values of $|x_2|$. \square

3.3 Optimality Conditions in the Convex Case

In the convex case we can establish necessary and sufficient optimality conditions. In particular we can state the following proposition.

Fig. 3.2 Saddle point

Proposition 3.8 (Optimality Conditions in the Convex Case) *Let $f : R^n \rightarrow R$ be continuously differentiable and convex on R^n . Then x^* is a global minimum point of f on R^n if and only if $\nabla f(x^*) = 0$. Moreover, if f is strictly convex on R^n and $\nabla f(x^*) = 0$, then x^* is the only stationary point and the unique global minimizer of f in R^n .*

Proof Necessity of the first assertion follows directly from Proposition 3.4, as x^* is obviously also a local minimizer. To show sufficiency we can note that the convexity of f implies that

$$f(x) \geq f(x^*) + \nabla f(x^*)^T(x - x^*)$$

for every pair $x, x^* \in R^n$, and hence, as $\nabla f(x^*) = 0$, we have $f(x) \geq f(x^*)$, for all $x \in R^n$. The last assertion follows from the first assertion and the definition of strict convexity. \square

A class of convex function of special interest in optimization is that of the convex quadratic functions. A quadratic function is a (infinitely differentiable) function of the form:

$$q(x) = \frac{1}{2}x^T Qx + c^T x,$$

where Q is a $n \times n$ symmetric matrix and $c \in R^n$. We have $\nabla q(x) = Qx + c$, and $\nabla^2 q(x) = Q$. We already know that q is convex if and only if the Hessian matrix Q is positive semidefinite and that q is strictly convex if and only if Q

is positive definite. We can state the following proposition, whose assertions are implicitly contained in the preceding results.

Proposition 3.9 (Minimization of a Quadratic Function) *Let $q(x) = \frac{1}{2}x^T Qx + c^T x$, with Q symmetric and $c \in R^n$. Then:*

- (a) *q has a global minimum if and only if Q is positive semidefinite and there exists x^* such that $Qx^* + c = 0$;*
- (b) *if Q is positive semidefinite then any stationary point is a global minimum point of q ;*
- (c) *q has a unique global minimum point if and only if Q is positive definite.*

Proof

Assertion (a). Suppose that $Qx^* + c = 0$ and that Q is positive semidefinite. The latter assumption implies that q is a convex function and hence, as $\nabla q(x^*) = 0$, by Proposition 3.8 it follows that x^* is an optimal solution.

Conversely, by Proposition 3.5, if q has a minimum point x^* , we have $\nabla q(x^*) = 0$ and the Hessian matrix $\nabla^2 q(x^*) = Q$ is positive semidefinite.

Assertion (b). If Q is semidefinite positive then q is convex and hence, again by Proposition 3.8 it follows that any stationary point is a global minimum of q .

Assertion (c). If Q is positive definite then q is strictly convex and the system $Qx + c = 0$ admits a unique solution x^* being Q nonsingular. Proposition 3.8 implies that x^* is the unique global minimum of q .

Conversely, assume that x^* is the unique minimum point. Then, assertion (a) implies that Q is positive semidefinite and the system $Qx + c = 0$ admits solution. From assertion (b) we have that any solution of the system $Qx + c = 0$ is a global minimum of q . Since the global minimum is unique, the matrix Q of the linear system must be nonsingular and hence positive definite. \square

Example 3.4 Consider the function $f : R^2 \rightarrow R$ defined by

$$f(x) = x_1^2 + x_2^2 + 2x_1x_2 - 2x_1 + 2x_2 = 0.$$

We have

$$\nabla f(x) = \begin{pmatrix} 2x_1 + 2x_2 - 2 \\ 2x_1 + 2x_2 + 2 \end{pmatrix} \quad \nabla^2 f(x) = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}.$$

It is easily seen that f is a quadratic convex function with positive semidefinite Hessian matrix. The equation $\nabla f(x) = 0$ has no solution and hence f does not admit minimum points. \square

An important class of problems with a convex quadratic objective function is that of *linear least squares problems*, already considered in Chap. 1, defined by:

$$\min f(x) = \frac{1}{2} \|Ax - b\|^2,$$

where A is a $(m \times n)$ real matrix and $b \in R^m$. We have

$$\nabla f(x) = A^T(Ax - b), \quad \nabla^2 f(x) = A^T A.$$

We know that this problem always has an optimal solution and then, as f is a convex quadratic function, it follows from Proposition 3.9 that the so called *normal equations*,

$$A^T A x = A^T b,$$

which express the condition $\nabla f(x) = 0$, must admit a solution and that every solution is an optimal global minimizer. In particular, if A has rank n , the matrix $A^T A$ is positive definite and hence non singular. Thus, we can define the *pseudoinverse* matrix

$$(A^T A)^{-1} A^T,$$

and the unique optimal solution is:

$$x^* = (A^T A)^{-1} A^T b.$$

Example 3.5 (Computation of the Regression Line) A well known example of least squares problems, is the *linear regression problem*, where we want to approximate the unknown functional relationship between an *independent variable* and a *dependent variable* with a linear model.

As an example, if we assume that the independent variable is the scalar $t \in R$, we want to determine the parameters $\alpha \in R$ and $\beta \in R$ that define the *regression line*

$$y(t) = \alpha t + \beta,$$

on the basis of a set of data pairs $\{(t_i, y_i), i = 1, \dots, m\}$ with distinct values t_i , obtained, for instance, through measurements and possibly affected by random errors.

Then the least squares problem is that of minimizing with respect to (α, β) the error function

$$f(\alpha, \beta) = \frac{1}{2} \sum_{i=1}^m (y_i - \alpha t_i - \beta)^2,$$

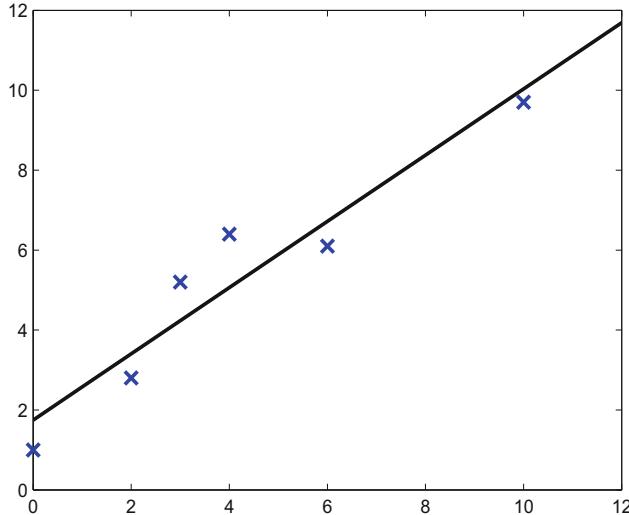


Fig. 3.3 Linear regression

which measures the distance between the measurements y_i and the model outputs $y(t_i) = \alpha t_i + \beta$, for $i = 1, \dots, m$ (Fig. 3.3).

The first order derivatives of f with respect to α and β are given by

$$\frac{\partial f}{\partial \alpha} = - \sum_{i=1}^m (y_i - \alpha t_i - \beta) t_i,$$

$$\frac{\partial f}{\partial \beta} = - \sum_{i=1}^m (y_i - \alpha t_i - \beta),$$

and the second order derivatives are

$$\frac{\partial^2 f}{\partial \alpha^2} = \sum_{i=1}^m t_i^2 \quad \frac{\partial^2 f}{\partial \alpha \partial \beta} = \sum_{i=1}^m t_i \quad \frac{\partial^2 f}{\partial \beta^2} = m.$$

By imposing the first order optimality condition we obtain

$$\alpha \sum_{i=1}^m t_i^2 + \beta \sum_{i=1}^m t_i = \sum_{i=1}^m y_i t_i,$$

$$\alpha \sum_{i=1}^m t_i + \beta m = \sum_{i=1}^m y_i.$$

Letting

$$\bar{t} = \frac{1}{m} \sum_{i=1}^m t_i \quad \bar{y} = \frac{1}{m} \sum_{i=1}^m y_i, \quad s_{tt} = \sum_{i=1}^m t_i^2 \quad s_{ty} = \sum_{i=1}^m t_i y_i,$$

we can write

$$\alpha s_{tt} + \beta m\bar{t} = s_{ty}, \quad \alpha \bar{t} + \beta = \bar{y}$$

whence it follows

$$\beta = \bar{y} - \alpha \bar{t}, \quad \alpha = \frac{s_{ty} - m\bar{t}\bar{y}}{s_{tt} - m\bar{t}^2}.$$

If we set

$$S_{ty} = s_{ty} - m\bar{t}\bar{y} = \sum_{i=1}^m (t_i - \bar{t})(y_i - \bar{y}), \quad S_{tt} = s_{tt} - m\bar{t}^2 = \sum_{i=1}^m (t_i - \bar{t})^2$$

we can express the optimal solution into the form

$$\alpha = \frac{S_{ty}}{S_{tt}}, \quad \beta = \bar{y} - \alpha \bar{t}.$$

Recalling the Schwarz inequality, and taking into account the fact that the numbers t_i are distinct, we have

$$s_{tt} - m\bar{t}^2 = \sum_{i=1}^m t_i^2 - \frac{1}{m} \left(\sum_{i=1}^m t_i \right)^2 > 0,$$

which proves that the Hessian matrix of f

$$\nabla^2 f(\alpha, \beta) = \begin{pmatrix} \sum_{i=1}^m t_i^2 & \sum_{i=1}^m t_i \\ \sum_{i=1}^m t_i & m \end{pmatrix},$$

is positive definite. □

3.4 Exercises

3.1 Let

$$f(x) = 1/2(x_1 - 1)^2 + 1/2(x_2 - 1)^2.$$

Define a descent direction $d \in R^2$ at the point $\bar{x} = (2, 2)^T$ such that $d \neq -\nabla f(\bar{x})$.

3.2 Consider the problem

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ x \in & R^n \end{aligned}$$

where $Q \in R^{n \times n}$ is a symmetric indefinite matrix, $c \in R^n$. Show that the problem does not admit minimum and maximum points.

3.3 Let $f : R^n \rightarrow R$ be a continuously differentiable function. Let x^* be a point such that

$$f(x^*) \leq f(x^* + \alpha d)$$

for every $\alpha \in R$ and $d \in R^n$. Show that $\nabla f(x^*) = 0$.

3.4 Let $f : R^n \rightarrow R$ be a continuously differentiable function and let $\bar{x} \in R^n$ be a point such that $\nabla f(\bar{x}) \neq 0$. Let $\{d_1, \dots, d_n\}$ be a basis for R^n . Prove that the set of directions

$$\{d_1, d_2, \dots, d_n, -d_1, -d_2, \dots, -d_n\}$$

contains at least a descent direction for f at \bar{x} .

3.5 Consider the problem

$$\begin{aligned} \min \quad & f(x) = \frac{1}{2}x^T Qx - b^T x \\ x \in & R^n, \end{aligned}$$

where Q is a symmetric positive semidefinite matrix $n \times n$. Suppose that the objective function f is bounded below. Prove that f admits a global minimum point.

Hint: any nonzero vector $b \in R^n$ can be uniquely decomposed as follows

$$b = b_{\mathcal{R}} + b_{\mathcal{N}},$$

where $b_{\mathcal{R}} \in \mathcal{R}(Q)$ and $b_{\mathcal{N}} \in \mathcal{N}(Q^T)$, being

$$\mathcal{R}(Q) = \{x \in R^n : Qy = x \quad y \in R^n\},$$

the (*range*) subspace of Q ,

$$\mathcal{N}(Q^T) = \{x \in R^n : Q^T x = 0\}$$

the *null* subspace of Q^T . Use this fact for proving that, under the assumption stated, there must exist a stationary point of the convex function f .

3.5 Notes and References

In this chapter we have stated the basic optimality conditions on local solutions of unconstrained problems, under the assumption that the objective function is (one or two times) continuously differentiable. The motivation was that the computational algorithms described in the sequel are essentially based on these assumptions. However, the local optimality conditions can also be established under weaker requirements, such as Gateaux differentiability. The interested reader is referred, for instance, to [200] and to [109]. Several tools for the recognition of the definiteness of the Hessian matrix, such as *Sylvester's condition*, are reported in [109].

Chapter 4

Optimality Conditions for Problems with Convex Feasible Set



We consider constrained optimization problems, where the feasible set is a convex set. First we introduce the concept of *feasible direction* and we state some elementary optimality conditions for general constrained optimization problems. In particular, we consider the case of linear constraints, where feasible directions can be easily characterized. Then we state optimality conditions for problems with convex feasible set, without imposing explicit regularity assumptions on the constraints that define the feasible set. We also particularize the optimality conditions to the case of box constraints. Finally, we introduce the concept of *projection on a convex set* and we give equivalent first-order optimality conditions based on the projection mapping.

4.1 Optimality Conditions Based on Feasible Directions

Let us consider the constrained optimization problems

$$\min f(x), \quad x \in S,$$

where $f : R^n \rightarrow R$ and $S \subset R^n$.

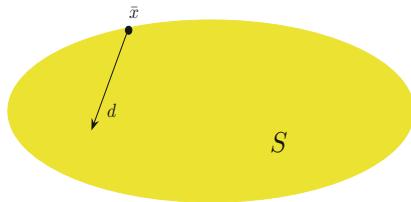
In the unconstrained case, i.e. when $S = R^n$, the concept of *descent direction* is fundamental for stating the optimality conditions and for designing unconstrained optimization algorithms.

In the constrained case, when $S \subset R^n$, in order to extend directly the conditions stated in the unconstrained case, we need also the concept of *feasible direction*.

This may yield useful conditions, provided that feasible directions exist and can be well characterized.

We state first a formal definition of feasible direction (Fig. 4.1).

Fig. 4.1 Feasible direction d at a given feasible point \bar{x}



Definition 4.1 (Feasible Direction) Let S be a subset of R^n and $x \in S$. A vector $d \in R^n$, $d \neq 0$ is a feasible direction for S at x if there exists a scalar $\bar{t} > 0$ such that

$$x + td \in S, \quad \text{for all } t \in [0, \bar{t}].$$

□

As a consequence of the above definition we can immediately state the following necessary optimality condition

Proposition 4.1 (Necessary Optimality Condition) *Let $x^* \in S$ be a local minimum point for the problem*

$$\min f(x), \quad x \in S;$$

then there cannot exist any feasible and descent direction at x^ .*

Proof Let us assume, by contradiction, that there exists a feasible and descent direction d at x^* . Then, in any neighborhood of x^* it is possible to find, for $t > 0$ and sufficiently small, a point $x^* + td \in S$ such that

$$f(x^* + td) < f(x^*),$$

and this contradicts the hypothesis that x^* is a local minimum point. □

If f is a differentiable function then, as we know, the condition

$$\nabla f(x)^T d < 0,$$

is sufficient to state that d is a descent direction for f at x , so that we get immediately the following result, which states that cannot exist a feasible direction d at x^* such that $\nabla f(x^*)^T d < 0$.

Proposition 4.2 (First Order Optimality Condition) *Let $f : R^n \rightarrow R$ be continuously differentiable in a neighborhood of a local minimum point $x^* \in S$ of the problem*

$$\min f(x), \quad x \in S.$$

Then we have that $\nabla f(x^)^T d \geq 0$, for every feasible direction $d \in R^n$ at x^* . \square*

By assuming that f is twice continuously differentiable and by using a known descent condition based on second order information, we can easily establish the following second order optimality conditions.

Proposition 4.3 (Second Order Optimality Condition) *Let $f : R^n \rightarrow R$ be twice continuously differentiable in a neighborhood of a local minimum point $x^* \in S$ of the problem*

$$\min f(x), \quad x \in S.$$

Then, for every feasible direction $d \in R^n$ at x^ we have:*

- (a) $\nabla f(x^*)^T d \geq 0$;
- (b) $d^T \nabla^2 f(x^*) d \geq 0$, if $\nabla f(x^*)^T d = 0$.

\square

4.2 The Case of Feasible Set Defined by Linear Constraints

Let us assume that the feasible set S is defined by a system of linear inequalities, i.e.

$$S = \{x \in R^n : Ax \leq b\},$$

where A is a $m \times n$ matrix and $b \in R^m$.

In this case we can give an explicit characterization of the feasible directions at a feasible point.

We denote by a_i^T the rows of A , with $i = 1, \dots, m$. Let x^* be a given feasible point, so that

$$a_i^T x^* \leq b_i \quad i = 1, \dots, m.$$

Let $I_0(x^*)$ be the index set of active constraints at x^* , i.e.,

$$I_0(x^*) = \{i : a_i^T x^* = b_i\}.$$

Example 4.1 Consider the following feasible set defined by $m = 3$ constraints and $n = 4$ variables:

$$\begin{aligned} 2x_1 - x_2 + 3x_3 + x_4 &\leq 4 \\ -x_1 + x_2 + 2x_3 + 3x_4 &\leq 5 \\ 2x_1 - 2x_2 + 6x_3 - x_4 &\leq 6 \end{aligned}$$

The point $x^* = (1 \ 1 \ 1 \ 0)^T$ is a feasible point where the first and the third inequalities constraints are satisfied as equalities. Then, the index set of active constraints is

$$I_0(x^*) = \{1, 3\}.$$

□

We can give the following characterization of feasible directions in case of linear constraints.

Proposition 4.4 (Feasible Directions: Linear Inequality Constraints) *Let x^* be such that $a_i^T x^* \leq b_i$ for $i = 1, \dots, m$. A vector $d \in R^n$, $d \neq 0$, is a feasible direction for S at x^* if and only if*

$$a_i^T d \leq 0 \quad \text{for all } i \in I_0(x^*). \tag{4.1}$$

Proof Let $d \in R^n$ be a feasible direction at x^* . Then, for $t > 0$ and t sufficiently small, we must have

$$a_i^T (x^* + td) = b_i + t a_i^T d \leq b_i \quad \text{for all } i \in I_0(x^*),$$

from which we get that (4.1) must hold.

Suppose now that (4.1) holds. This implies that the constraints active at x^* are always satisfied at every point $x^* + td$ with $t > 0$. To prove sufficiency of (4.1), we must show that also the inactive constraints are satisfied at every point $x^* + td$ with $t > 0$ for t sufficiently small.

Since $a_i^T x^* < b_i$ for $i \notin I_0(x^*)$, taking t sufficiently small we have

$$a_i^T(x^* + td) \leq b_i \quad \text{for all } i \notin I_0(x^*),$$

so that we can conclude that there exists a number $\bar{t} > 0$ such that

$$A(x^* + td) \leq b,$$

for all $t \in [0, \bar{t}]$, i.e., that d is a feasible direction for S at x^* . \square

Now let us consider a feasible set S defined by linear inequality and equality constraints, i.e.,

$$S = \{x \in R^n : Ax \leq b, Gx = c\},$$

where A is a $m \times n$ matrix with rows $a_i^T, i = 1, \dots, p, b \in R^m$, G is a $p \times n$ matrix with rows $g_j^T, j = 1, \dots, p$, and $c \in R^p$. If we rewrite the constraints defining S as inequalities, that is

$$\begin{aligned} a_i^T x &\leq b_i & i = 1, \dots, m \\ g_j^T x &\leq c_j & j = 1, \dots, p \\ -g_j^T x &\leq -c_j & j = 1, \dots, p, \end{aligned} \tag{4.2}$$

we can observe that the last $2p$ constraints are active in correspondence to any feasible point x^* . As regards the first m constraints, we denote by $I_0(x^*)$ the index set of active constraints, i.e.,

$$I_0(x^*) = \{i : a_i^T x^* = b_i\}.$$

From Proposition 4.4 we get the following result.

Proposition 4.5 (Feasible Directions for Linear Constraints) *Let x^* be such that $a_i^T x^* \leq b_i$ for $i = 1, \dots, m$, $g_j^T x^* = c_j$ for $j = 1, \dots, p$. A vector $d \in R^n$, $d \neq 0$, is a feasible direction for S at x^* , if and only if*

$$\begin{aligned} a_i^T d &\leq 0 & \text{for all } i \in I_0(x^*) \\ g_j^T d &= 0 & \text{for all } j = 1, \dots, p. \end{aligned} \tag{4.3}$$

Now let us consider the problem

$$\begin{aligned} & \min f(x) \\ & Ax \leq b \\ & Gx = c \end{aligned} \tag{4.4}$$

where f is continuously differentiable.

Let x^* be a local minimum point. Then x^* is a feasible point, i.e., $Ax^* - b \leq 0$, $Gx^* = c$. Furthermore, from Propositions 4.2 and 4.4 we get that there cannot exist $d \in R^n$ such that

$$\begin{aligned} & \nabla f(x^*)^T d < 0 \\ & a_i^T d \leq 0 \quad i \in I_0(x^*) \\ & g_j^T d = 0 \quad j = 1, \dots, p. \end{aligned} \tag{4.5}$$

In the case of system (4.5) it is possible to use the theorems of the alternative, to obtain a special case of the Karush-Kuhn-Tucker (KKT) optimality conditions established for general nonlinear programming problems. The study of the KKT conditions will be performed in Chap. 5, following a different approach, not based on theorems of the alternative.

Some essential results on the theorems of alternative and the application to derive optimality conditions in linear and nonlinear programming will be given in Chap. 7.

4.2.1 Feasible Directions for Problems with Box Constraints

Let us consider a feasible set defined by box constraints, i.e.

$$S = \{x \in R^n : l_i \leq x_i \leq u_i, i = 1, \dots, n\}$$

with $l_i < u_i$ for $i = 1, \dots, n$. From Proposition 4.4 we get immediately the following result.

Proposition 4.6 (Feasible directions for Box Constraints) *Let $\bar{x} \in S$. A vector $d \in R^n$, $d \neq 0$, is a feasible direction for S at \bar{x} , if and only if*

$$\begin{aligned} d_i &\geq 0 && \text{for all } i : \bar{x}_i = l_i \\ d_i &\leq 0 && \text{for all } i : \bar{x}_i = u_i \end{aligned} \tag{4.6}$$

□

Given a feasible point \bar{x} , we can prove a result on the set $D(\bar{x})$ of feasible directions at \bar{x} that will be used later to state optimality conditions. We denote by C the set of the coordinate directions and their opposite, i.e.,

$$C = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\}$$

Let

$$D_c(\bar{x}) = C \cap D(\bar{x}),$$

i.e., $D_c(\bar{x})$ is the subset of C done by feasible directions at \bar{x} .

Example 4.2 Consider a feasible set defined in R^3 by the following box constraints:

$$\begin{aligned} -1 &\leq x_1 \leq 10 \\ -2 &\leq x_2 \leq -1 \\ 0 &\leq x_3 \leq 2 \end{aligned}$$

and let $\bar{x} = (0 \ -2 \ 2)^T$. We have

$$D_c(\bar{x}) = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \right\}$$

□

Given a set $Y = \{y^1, y^2, \dots, y^m\} \subset R^n$ we indicate by $cone(Y)$ the convex cone of Y , i.e.,

$$cone(Y) = \{y \in R^n : y = \sum_{i=1}^m \beta_i y^i, \ \beta_i \geq 0, \ i = 1, \dots, m\}$$

Proposition 4.7 *Let $\bar{x} \in S$,*

$$C = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\},$$

and

$$D_c(\bar{x}) = C \cap D(\bar{x}).$$

(continued)

Proposition 4.7 (continued)
Then for any $d \in D(\bar{x})$ we have

$$d \in \text{cone}(D_c(\bar{x})).$$

Proof Let $d \in D(\bar{x})$ and let

$$I^+ = \{i \in \{1, \dots, n\} : d_i > 0\} \quad I^- = \{i \in \{1, \dots, n\} : d_i < 0\}$$

From Proposition 4.6 we have

- (a) $i \in I^+$ implies $\bar{x}_i < u_i$;
- (b) $i \in I^-$ implies $\bar{x}_i > l_i$.

We can write

$$d = \sum_{i \in I^+} d_i e_i + \sum_{i \in I^-} |d_i|(-e_i)$$

and hence the thesis is proved taking into account that, from (a) and (b), it follows that $e_i \in D_c(\bar{x})$ for every $i \in I^+$, and $-e_i \in D_c(\bar{x})$ for every $i \in I^-$. \square

4.3 Optimality Conditions over a Convex Set

Let us consider an optimization problem whose feasible set S is a *convex set*. Thanks to the convexity of S it is possible to define a feasible direction at a given point by using any different feasible point.

Proposition 4.8 (Feasible Directions of a Convex Set) *Let $S \subseteq R^n$ be a convex set and let $\bar{x} \in S$. Then, if $S \neq \{\bar{x}\}$, for every $x \in S$ such that $x \neq \bar{x}$, the direction $d = x - \bar{x}$ is a feasible direction for S at \bar{x} .*

Proof Let $\bar{x} \in S$. For every $x \in S$ such that $x \neq \bar{x}$, thanks to the convexity of S , we have that $(1-t)\bar{x} + tx \in S$ for all $t \in [0, 1]$ and hence $\bar{x} + t(x - \bar{x}) \in S$ for all $t \in [0, 1]$. It follows that $d = (x - \bar{x}) \neq 0$ is a feasible direction for S at \bar{x} . \square

By assuming that f is (twice) continuously differentiable we can state the following (second) first order optimality conditions.

Proposition 4.9 (Optimality Conditions) *Let $x^* \in S$ be a local minimum point of the problem*

$$\min f(x), \quad x \in S,$$

where $S \subseteq \mathbb{R}^n$ is a convex set. Suppose that f is continuously differentiable in a neighborhood of x^ . Then we have*

$$\nabla f(x^*)^T(x - x^*) \geq 0, \quad \text{for all } x \in S. \quad (4.7)$$

Moreover, if f is twice continuously differentiable in a neighborhood of x^ then we have also*

$$(x - x^*)^T \nabla^2 f(x^*)(x - x^*) \geq 0, \quad (4.8)$$

for all $x \in S$ such that $\nabla f(x^)^T(x - x^*) = 0$.*

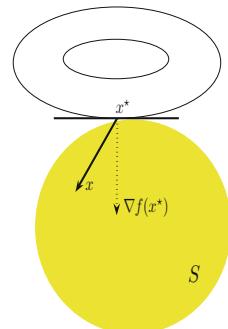
Proof If $S = \{x^*\}$ the thesis is true. Then, assume there exists $x \in S$ such that $x \neq x^*$. From Proposition 4.8 it follows that $d = (x - x^*)$ is a feasible direction for S at x^* . Then the assertion follows immediately from Propositions 4.2 and 4.3. \square

The geometric interpretation of condition (4.7) is shown in Fig. 4.2, where it is assumed that $x^* \in S$ is a local minimum point. We can observe that for all $x \in S$ the direction $d = x - x^*$ (which is a feasible direction thanks to the convexity of S) can not make an acute angle with $-\nabla f(x^*)$ (otherwise it would be a descent direction).

We say that $x^* \in S$ is a *critical point* if the necessary condition (4.7) of Proposition 4.9 holds at x^* .

It can be easily shown that condition (4.7) is a *necessary and sufficient condition of global minimum point* if we assume that f is convex.

Fig. 4.2 Geometric interpretation of the optimality condition



Proposition 4.10 (Condition of Global Minimum Point: The Convex Case) Let S be a convex subset of R^n and suppose that f is a convex and continuously differentiable function on an open set containing S . Then $x^* \in S$ is a global minimum point of the problem

$$\min f(x), \quad x \in S$$

if and only if

$$\nabla f(x^*)^T(x - x^*) \geq 0, \quad \text{for all } x \in S.$$

Proof Proposition 4.9 states that (4.7) is a necessary condition. Since f is convex and $x^* \in S$, using a known result on convex functions, we have that for every $x \in S$ we can write

$$f(x) \geq f(x^*) + \nabla f(x^*)^T(x - x^*).$$

If (4.7) holds then we have $f(x) \geq f(x^*)$ for every $x \in S$, and this implies that x^* is a global minimum point. \square

We can particularize the optimality conditions to the case of problems with box constraints. In particular we can state the following result.

Proposition 4.11 (Necessary Optimality Condition for Problems with Box Constraints) Let S be a convex subset of R^n defined by box constraints, i.e.,

$$S = \{x \in R^n : l \leq x \leq u\},$$

and suppose that f is continuously differentiable function on an open set containing S . Then $x^* \in S$ is a critical point of the problem

$$\min f(x), \quad x \in S,$$

i.e., satisfies the necessary condition (4.7) of Proposition 4.9, if and only if, for $i = 1, \dots, n$, we have

$$\frac{\partial f(x^*)}{\partial x_i} \begin{cases} \geq 0 & \text{if } x_i^* = l_i \\ = 0 & \text{if } l_i < x_i^* < u_i \\ \leq 0 & \text{if } x_i^* = u_i \end{cases} \quad (4.9)$$

Proof The convexity of the feasible set implies that a feasible point x^* is a critical point if and only if

$$\nabla f(x^*)^T d \geq 0 \quad \text{for every } d \in D(x^*). \quad (4.10)$$

First we prove that (4.10) implies (4.9). Let $i \in \{1, \dots, n\}$ such that $x_i^* = l_i$ and consider the direction $d^+ = e_i$, where e_i is the i -th coordinate direction. Proposition 4.6 implies that the direction d^+ is a feasible direction, and hence, from (4.10) it follows

$$\nabla f(x^*)^T d^+ = \frac{\partial f(x^*)}{\partial x_i} \geq 0.$$

In a similar way, let $i \in \{1, \dots, n\}$ such that $x_i^* = u_i$ and consider the direction $d^- = -e_i$. Using again Proposition 4.6 we have that d^- is a feasible direction, and hence, from (4.10) it follows

$$\frac{\partial f(x^*)}{\partial x_i} \leq 0.$$

Finally, let $i \in \{1, \dots, n\}$ such that $l_i < x_i^* < u_i$. The directions $d^+ = e_i, d^- = -e_i$ are feasible directions, and hence, from (4.10) it follows

$$\nabla f(x^*)^T d^+ = \frac{\partial f(x^*)}{\partial x_i} \geq 0 \quad \nabla f(x^*)^T d^- = -\frac{\partial f(x^*)}{\partial x_i} \geq 0,$$

from which we obtain

$$\frac{\partial f(x^*)}{\partial x_i} = 0.$$

Now we prove that (4.9) implies (4.10). Consider the set C of the coordinate directions and their opposite, i.e.,

$$C = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\},$$

and let

$$D_c(x^*) = C \cap D(x^*),$$

i.e., $D_c(\bar{x})$ is the subset of C done by feasible directions at x^* . From (4.9) it follows

$$\nabla f(x^*)^T d \geq 0 \quad \text{for every } d \in D_c(x^*). \quad (4.11)$$

Indeed, suppose that e_i is a feasible direction at x^* , i.e., $e_i \in D_c(x^*)$. From (4.6) it follows $x_i^* < u_i$ and hence, using (4.9), we have $\frac{\partial f(x^*)}{\partial x_i} \geq 0$, so that we can write

$$\nabla f(x^*)^T d = \nabla f(x^*)^T e_i = \frac{\partial f(x^*)}{\partial x_i} \geq 0$$

In a similar way, the same conclusion can be obtained assuming that $-e_i$ is a feasible direction at x^* . Therefore, from (4.11) and Proposition 4.7, we can conclude that (4.10) holds. \square

From Propositions 4.11 and 4.10 we get immediately the following result.

Proposition 4.12 (Condition of Global Minimum Point for Problems with Box Constraints: The Convex Case) *Let S be a convex subset of R^n defined by box constraints, i.e.,*

$$S = \{x \in R^n : l \leq x \leq u\},$$

and suppose that f is a convex and continuously differentiable function on an open set containing S . Then $x^ \in S$ is a global minimum point of the problem*

$$\min f(x), \quad x \in S$$

if and only if we have for $i = 1, \dots, n$

$$\frac{\partial f(x^*)}{\partial x_i} \begin{cases} \geq 0 & \text{if } x_i^* = l_i \\ = 0 & \text{if } l_i < x_i^* < u_i \\ \leq 0 & \text{if } x_i^* = u_i \end{cases} \quad (4.12)$$

4.4 Projection on a Convex Set

Let $S \subseteq R^n$ be a closed, nonempty convex set and let $x \in R^n$ be a given point. Consider the problem

$$\min \|x - y\|, \quad y \in S, \quad (4.13)$$

that is, the problem of determining the point in S which is at minimum distance from the point x .

Since the objective function is coercive on the closed and nonempty set S , there exist nonempty and compact level sets, and hence the above problem admits

solution. Moreover, as the Euclidean norm is a strictly convex function we have the solution is unique. Then we can state the following definition.

Definition 4.2 (Projection of a Point on a Convex Set) Let $S \subseteq R^n$ be a closed, nonempty convex set and let $x \in R^n$ be a given point. The projection of x on S is the solution $p(x)$ of the following problem

$$\min \|x - y\|, \quad y \in S,$$

that is, $p(x) \in S$ is the point such that

$$\|x - p(x)\| \leq \|x - y\|, \quad \text{for all } y \in S.$$

□

The next result concerns the projection of x on S .

Proposition 4.13 (Characterization of the Projection) Let $S \subseteq R^n$ be a convex, nonempty set, let $x \in R^n$ be a given point and let $\|\cdot\|$ be the Euclidean norm. Then

- (i) a point $y^* \in S$ is the projection of x on S , i.e., $y^* = p(x)$ if and only if

$$(x - y^*)^T(y - y^*) \leq 0 \quad \text{for all } y \in S;$$

- (ii) the projection mapping is continuous and non-expansive, that is

$$\|p(x) - p(z)\| \leq \|x - z\| \quad \text{for all } x, z \in R^n.$$

Proof The projection $p(x)$ of x is the unique solution of the problem

$$\min f(y) \equiv \frac{1}{2} \|x - y\|^2, \quad y \in S,$$

(which is equivalent to problem (4.13)). Therefore, from Proposition 4.10 it follows that $y^* = p(x)$, if and only if $\nabla f(y^*)^T(y - y^*) \geq 0$ for every $y \in S$, that is, if and only if $-(x - y^*)^T(y - y^*) \geq 0$, for every $y \in S$, and this proves (i).

If $x, z \in \mathbb{R}^n$ are given points, denoting by $p(x), p(z) \in S$ the projections, from (i), applied to the points x and z , we must have

$$(x - p(x))^T(y - p(x)) \leq 0, \quad \text{for all } y \in S$$

$$(z - p(z))^T(y - p(z)) \leq 0, \quad \text{for all } y \in S.$$

Therefore, (since $p(x), p(z)$ are particular points of S), we can write, setting $y = p(z)$ in the first inequality and $y = p(x)$ in the second one:

$$(x - p(x))^T(p(z) - p(x)) \leq 0$$

$$(z - p(z))^T(p(x) - p(z)) \leq 0.$$

Summing the two inequalities we obtain

$$(x - p(x) - z + p(z))^T(p(z) - p(x)) \leq 0,$$

from which it follows

$$\|p(z) - p(x)\|^2 - (p(z) - p(x))^T(z - x) \leq 0,$$

Using the Cauchy-Schwarz inequality we obtain

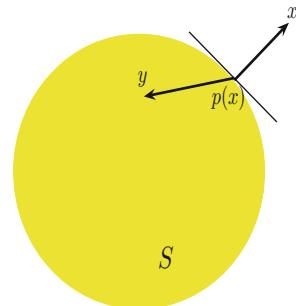
$$\|p(z) - p(x)\|^2 \leq (p(z) - p(x))^T(z - x) \leq \|p(z) - p(x)\|\|z - x\|,$$

and we can conclude that (ii) holds. \square

The geometric interpretation of the result of Proposition 4.13 is shown in Fig. 4.3. The vector $x - p(x)$ must make an angle $\geq \pi/2$, with every vector $y - p(x)$, where y varies within the set S .

The projection mapping requires to solve an optimization problem. In general, to compute the projection of a vector on a set, an iterative method must be applied.

Fig. 4.3 Geometric interpretation of the projection $p(x)$



In simple cases, like the next two cases, the projection can be determined in closed form.

Example 4.3 (Projection on a Set Defined by Nonnegative Constraints) Consider the set

$$S = \{y \in R^n : y_i \geq 0, i = 1, \dots, n\}.$$

Let \bar{x} be a point in R^n ; the projection y^* of \bar{x} on S can be evaluated by minimizing the function

$$f(y) \equiv \sum_{i=1}^n (y_i - \bar{x}_i)^2, \quad y_i \geq 0, \quad i = 1, \dots, n.$$

This problem can be obviously decomposed into n independent problems of the form

$$\min (y_i - \bar{x}_i)^2, \quad y_i \geq 0.$$

Then, it is easily seen that the i -th component y_i^* of the solution $y^* \in S$ is given by:

$$y_i^* = \begin{cases} \bar{x}_i & \text{if } \bar{x}_i \geq 0 \\ 0 & \text{if } \bar{x}_i < 0 \end{cases}$$

It can be verified that y^* satisfies the necessary and sufficient condition (i) of Proposition 4.13, i.e., the following condition

$$(\bar{x} - y^*)^T (y - y^*) \leq 0 \quad \text{for all } y \in S.$$

From the definition of y^* it follows

$$\begin{aligned} \bar{x}_i - y_i^* &= 0 && \text{if } \bar{x}_i \geq 0 \\ \bar{x}_i - y_i^* &= \bar{x}_i && \text{if } \bar{x}_i < 0, \end{aligned}$$

and hence, for every $y \in S$ we can write

$$(\bar{x} - y^*)^T (y - y^*) = \sum_{i:\bar{x}_i < 0} \bar{x}_i y_i \leq 0.$$

□

Example 4.4 (Projection on a Set Defined by Box Constraints) Consider the set

$$S = \{y \in R^n : l \leq y \leq u\}.$$

Let \bar{x} be a point in R^n . Then, reasoning as in the preceding case we can easily establish that the projection y^* of \bar{x} on S has components y_i^* , with $i = 1, \dots, n$, defined by:

$$y_i^* = \begin{cases} l_i & \text{if } \bar{x}_i < l_i \\ \bar{x}_i & \text{if } l_i \leq \bar{x}_i \leq u_i \\ u_i & \text{if } \bar{x}_i > u_i \end{cases}$$

We can show that y^* satisfies the necessary and sufficient condition (i) of Proposition 4.13. Recalling the definition of y^* we can write

$$(\bar{x} - y^*)^T (y - y^*) = \sum_{i:\bar{x}_i < l_i} (\bar{x}_i - l_i) (y_i - l_i) + \sum_{i:\bar{x}_i > u_i} (\bar{x}_i - u_i) (y_i - u_i) \leq 0,$$

being $(y_i - l_i) \geq 0$, $(y_i - u_i) \leq 0$. □

The projection mapping allows us to state a necessary optimality condition which is equivalent to condition (4.7) of Proposition 4.9, that we used to give the definition of critical point.

Proposition 4.14 *Consider the problem*

$$\min f(x), \quad x \in S,$$

where $S \subseteq R^n$ is a convex set. Let $x^* \in S$ and suppose that f is continuously differentiable in a neighborhood of x^* . Then the point x^* is a critical point if and only if

$$x^* = p[x^* - s \nabla f(x^*)], \quad (4.14)$$

where s is any positive number.

Proof Proposition 4.13 implies that the point x^* is the projection of the point $x^* - s \nabla f(x^*)$ if and only if

$$(x^* - s \nabla f(x^*) - x^*)^T (x - x^*) \leq 0, \quad \text{for all } x \in S,$$

that is, if and only if

$$\nabla f(x^*)^T (x - x^*) \geq 0, \quad \text{for all } x \in S,$$

that is, if and only if x^* is a critical point. □

From the above proposition and Proposition 4.9 we get immediately the following optimality condition.

Proposition 4.15 (Optimality Condition) *Let $x^* \in S$ be a local minimum point of problem*

$$\min f(x), \quad x \in S$$

where $S \subseteq R^n$ is a convex set and suppose that f is continuously differentiable in a neighborhood of x^ . Then we have*

$$x^* = p[x^* - s \nabla f(x^*)], \quad (4.15)$$

where s is any positive number. □

It can be easily verified that under the convexity assumption on f condition (4.14) is a necessary and sufficient condition of global minimum.

Proposition 4.16 (Condition of Global Minimum Point: The Convex Case) *Let S be a convex subset of R^n and assume that f is a convex, continuously differentiable function over an open set containing S . Then $x^* \in S$ is a global minimum point of problem*

$$\min f(x), \quad x \in S$$

if and only if

$$x^* = p[x^* - s \nabla f(x^*)],$$

where s is a positive number.

Proof The proof follows from Propositions 4.10 and 4.14. □

4.5 Exercises

4.1 Consider the problem:

$$\min 2x_1^2 - 2x_2 + 3x_3$$

$$x_1 - 2x_2 + x_3 = 0.$$

Define a feasible and descent direction for the objective function at $\bar{x} = (0 \ 0 \ 0)^T$.

4.2 Consider the problem with box constraints:

$$\min \frac{1}{2}cx_1^2 + \frac{1}{2}x_1^2x_2^2 - \frac{1}{3}x_2^3$$

$$1 \leq x_1 \leq 2.$$

Determine the interval of values of parameter c for which the point $\bar{x} = (1 \ 1)^T$ satisfies the necessary optimality conditions.

4.3 Consider the problem with box constraints:

$$\min f(x)$$

$$l \leq x \leq u,$$

where $f : R^n \rightarrow R$ is a convex continuously differentiable function. Let \bar{x} a feasible point such that $\bar{x}_n = u_n$ and let

$$\nabla f(\bar{x}) = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Prove that \bar{x} is a global minimum point.

4.6 Notes and References

The chapter introduces basic concepts (i.e., feasible direction, projection on a convex set) and optimality conditions for optimization problems whose feasible set is convex. The matter of this chapter is used later (see Chap. 20) for presenting algorithms for constrained optimization and is mainly based on classical books like [12] and [16].

Chapter 5

Optimality Conditions for Nonlinear Programming



In this chapter we consider finite dimensional Nonlinear Programming problems with a finite number of equality and inequality constraints and we state the basic first and second order optimality conditions. Using the penalty technique introduced by McShane in [185], we start by establishing the first order necessary conditions known as *Fritz John conditions*. Then, under suitable assumptions on the constraints, known as *constraint qualifications*, we obtain the *Karush-Kuhn-Tucker* (KKT) first order necessary conditions and we show that in the convex case these conditions are also sufficient for global optimality. Subsequently we give some basic result on second order optimality conditions and finally we specialize our results to some important classes of linearly constrained problems.

In Chap. 7 we give a different derivation of the optimality conditions, based on the concept of *tangent directions* and on the theorems of the alternative.

5.1 Problem Formulation and Notation

We consider problems of the form

$$\begin{aligned} & \min f(x) \\ & h(x) = 0 \quad g(x) \leq 0, \end{aligned} \tag{5.1}$$

where $f : R^n \rightarrow R$, $h : R^n \rightarrow R^p$ and $g : R^n \rightarrow R^m$ are assumed to be continuously differentiable (at least once in the study of first order conditions and twice in the case of second order conditions) on some open neighborhood of the region of interest.

We denote by $S = \{x \in R^n : h(x) = 0, g(x) \leq 0\}$ the feasible set of Problem (5.1), which is assumed to be non empty, and we indicate by $I_0(x^*)$ the index set of the inequality constraints that are active at $x^* \in S$, that is:

$$I_0(x^*) = \{i : g_i(x^*) = 0\}.$$

5.2 Fritz John Conditions

Fritz John (FJ) conditions are among the first optimality conditions introduced in Nonlinear Programming as an extension of the *Lagrange multiplier rule* given for equality constrained problems. FJ conditions are expressed with reference to an auxiliary *Lagrangian function* of the form.

$$L(x, \lambda_0, \lambda, \mu) = \lambda_0 f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \mu_i h_i(x),$$

where the auxiliary real variables $\lambda_0, \lambda_i, i = 1, \dots, m, \mu_i, i = 1, \dots, p$ are viewed as (*generalized*) *Lagrange multipliers* and we have set:

$$\lambda = (\lambda_1, \dots, \lambda_m), \quad \mu = (\mu_1, \dots, \mu_p).$$

We establish FJ conditions in the following proposition, where we follow essentially the formulation given in [16]. However, we refer to the auxiliary penalty function introduced in [109], which will be useful in the study of second order conditions.

Proposition 5.1 (Fritz John First Order Necessary Conditions) *Let x^* be a local minimum point of Problem (5.1) and suppose that the functions f, g, h are continuously differentiable in an open neighborhood of x^* .*

Then there exist multipliers $\lambda_0^ \in R, \lambda^* \in R^m, \mu^* \in R^p$ such that:*

(a) *the following conditions (Fritz John conditions) hold*

$$\begin{aligned} \lambda_0^* \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla h_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0, \quad i = 1, \dots, m \\ (\lambda_0^*, \lambda^*) &\geq 0, \quad (\lambda_0^*, \lambda^*, \mu^*) \neq 0, \\ g(x^*) &\leq 0, \quad h(x^*) = 0; \end{aligned} \tag{5.2}$$

(continued)

Proposition 5.1 (continued)

(b) *in every neighborhood of x^* there exists a point x such that*

$$\lambda_i^* g_i(x) > 0, \quad \text{for every } i \text{ such that } \lambda_i^* > 0,$$

$$\mu_i^* h_i(x) > 0, \quad \text{for every } i \text{ such that } \mu_i^* \neq 0.$$

Proof Preliminarily we observe that, as x^* is a local minimizer of Problem (5.1), then it must be feasible and hence $g(x^*) \leq 0$ and $h(x^*) = 0$. Moreover, there must exist a closed sphere $\bar{B}(x^*, \varepsilon) = \{x : \|x - x^*\| \leq \varepsilon\}$ with $\varepsilon > 0$, such that

$$f(x) \geq f(x^*) \quad \text{for all } x \in \bar{B}(x^*, \varepsilon) \text{ such that } g(x) \leq 0, h(x) = 0. \quad (5.3)$$

As the m functions g_i are continuous, we can choose ε sufficiently small to have that all the inequality constraints which are not active at x^* are negative in the sphere considered, that is

$$g_i(x) < 0, \quad \text{for all } x \in \bar{B}(x^*, \varepsilon) \text{ and every } i \text{ such that } g_i(x^*) < 0. \quad (5.4)$$

We can also assume that ε is sufficiently small to have that the differentiability assumptions hold on the neighborhood considered.

Now, for every fixed integer $k > 0$, let us define the following auxiliary function F_k containing penalty terms on the constraints:

$$F_k(x) = f(x) + \frac{1}{4} \|x - x^*\|^4 + \frac{k}{3} \sum_{i=1}^m (g_i^+(x))^3 + \frac{k}{2} \sum_{i=1}^p h_i^2(x),$$

where $g_i^+(x) = \max\{g_i(x), 0\}$. We can easily verify that $(g_i^+(x))^3$ is a continuously differentiable function with gradient

$$\nabla(g_i^+(x))^3 = 3(g_i^+(x))^2 \nabla g_i(x)$$

Let us consider the constrained problem

$$\begin{aligned} & \min F_k(x) \\ & x \in \bar{B}(x^*, \varepsilon) \end{aligned} \quad (5.5)$$

As F_k is continuous and $\bar{B}(x^*, \varepsilon)$ is compact, by Weierstrass theorem we have that, for every k there exists a minimum point $x_k \in \bar{B}(x^*, \varepsilon)$ of problem (5.5). Moreover, as $x^* \in \bar{B}(x^*, \varepsilon)$ we must have, for every k :

$$F_k(x_k) \leq F_k(x^*) = f(x^*), \quad (5.6)$$

where the last equality follows from the assumption that $x^* \in S$, which implies $g_i^+(x^*) = 0$ for every $i = 1, \dots, m$ and $h_i(x^*) = 0$ for every $i = 1, \dots, p$.

As $\bar{B}(x^*, \varepsilon)$ is compact, the sequence $\{x_k\}$ obtained by letting $k = 1, 2, \dots$ must admit a subsequence (which we relabel again $\{x_k\}$) converging to some $\bar{x} \in \bar{B}(x^*, \varepsilon)$. Thus, recalling (5.6) and the expression of $F_k(x)$, we can write

$$\frac{1}{3} \sum_{i=1}^m (g_i^+(x_k))^3 + \frac{1}{2} \sum_{i=1}^p h_i^2(x_k) \leq \frac{1}{k} \left(f(x^*) - f(x_k) - \frac{1}{4} \|x_k - x^*\|^4 \right).$$

As the sequence is bounded, taking limits for $k \rightarrow \infty$ we have

$$\lim_{k \rightarrow \infty} \left(\frac{1}{3} \sum_{i=1}^m (g_i^+(x_k))^3 + \frac{1}{2} \sum_{i=1}^p h_i^2(x_k) \right) = 0,$$

and hence the continuity of g_i and h_i , implies $g(\bar{x}) \leq 0$ $h(\bar{x}) = 0$.

From (5.6), recalling the expression of $F_k(x)$ it follows also that, for all k , we have

$$f(x_k) + \frac{1}{4} \|x_k - x^*\|^4 \leq f(x^*),$$

and hence we get, in the limit:

$$f(\bar{x}) + \frac{1}{4} \|\bar{x} - x^*\|^4 \leq f(x^*).$$

On the other hand, as \bar{x} is feasible, by (5.3) we have $f(\bar{x}) \geq f(x^*)$, so that

$$f(x^*) + \frac{1}{4} \|\bar{x} - x^*\|^4 \leq f(\bar{x}) + \frac{1}{4} \|\bar{x} - x^*\|^4 \leq f(x^*),$$

which implies $\bar{x} = x^*$. Thus we can assert that the sequence considered converges to x^* and hence, for sufficiently large values of k , the point x_k must be an unconstrained

minimum point of F_k in the interior of the sphere. Therefore the first order necessary optimality condition must be satisfied and we have

$$\begin{aligned} \nabla f(x_k) + \|x_k - x^*\|^2(x_k - x^*) + \sum_{i=1}^m k(g_i^+(x_k))^2 \nabla g_i(x_k) \\ + \sum_{i=1}^p kh_i(x_k) \nabla h_i(x_k) = 0. \end{aligned} \quad (5.7)$$

Now, for every k , let us define the numbers

$$\begin{aligned} w^k &= \left(1 + \sum_{i=1}^m [k(g_i^+(x_k))^2]^2 + \sum_{i=1}^p (kh_i(x_k))^2 \right)^{1/2}, \\ \lambda_0^k &= \frac{1}{w^k}, \quad \lambda_i^k = \frac{k(g_i^+(x_k))^2}{w^k}, \quad i = 1, \dots, m \\ \mu_i^k &= \frac{kh_i(x_k)}{w^k}, \quad i = 1, \dots, p. \end{aligned}$$

Letting $\lambda^k = (\lambda_1^k, \dots, \lambda_m^k)$ and $\mu^k = (\mu_1^k, \dots, \mu_p^k)$, and defining the vector $(\lambda_0^k, \lambda^k, \mu^k)$, we can easily verify that this vector has unit norm, as

$$\begin{aligned} \|(\lambda_0^k, \lambda^k, \mu^k)\|^2 &= (\lambda_0^k)^2 + \sum_{i=1}^m (\lambda_i^k)^2 + \sum_{i=1}^p (\mu_i^k)^2 \\ &= \left(\frac{1}{w^k} \right)^2 + \sum_{i=1}^m \left(\frac{k(g_i^+(x_k))^2}{w^k} \right)^2 + \sum_{i=1}^p \left(\frac{kh_i(x_k)}{w^k} \right)^2 \\ &= \left(\frac{1}{w^k} \right)^2 \left(1 + \sum_{i=1}^m (k(g_i^+(x_k))^2)^2 + \sum_{i=1}^p (kh_i(x_k))^2 \right) \\ &= \left(\frac{w^k}{w^k} \right)^2 = 1. \end{aligned}$$

It follows that the points $(\lambda_0^k, \lambda^k, \mu^k)$ remain within the unit closed sphere and hence we can extract a subsequence (which we relabel again $\{(\lambda_0^k, \lambda^k, \mu^k)\}$), converging to some vector $(\lambda_0^*, \lambda^*, \mu^*)$ with unit norm. Thus we can write

$$\lim_{k \rightarrow \infty} \lambda_0^k = \lambda_0^*, \quad \lim_{k \rightarrow \infty} \lambda_i^k = \lambda_i^* \quad \lim_{k \rightarrow \infty} \mu_i^k = \mu_i^*,$$

where

$$\|(\lambda_0^*, \lambda^*, \mu^*)\| = 1. \quad (5.8)$$

Now, for every fixed k , dividing both members of (5.7) by $w^k > 0$, we get

$$\lambda_0^k \nabla f(x_k) + \frac{\|x_k - x^*\|^2(x_k - x^*)}{w^k} + \sum_{i=1}^m \lambda_i^k \nabla g_i(x_k) + \sum_{i=1}^p \mu_i^k \nabla h_i(x_k) = 0. \quad (5.9)$$

Taking limits for $k \rightarrow \infty$ and recalling that x_k converges to x^* , we get

$$\lambda_0^* \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla h_i(x^*) = 0. \quad (5.10)$$

As $\lambda_0^k \geq 0$ and $\lambda_i^k \geq 0$, for $i = 1, \dots, m$ we must have, in the limit for $k \rightarrow \infty$

$$\lambda_0^* \geq 0, \quad \lambda^* \geq 0. \quad (5.11)$$

Moreover, as x_k converges to x^* , recalling (5.4), we have that $g_i(x_k) < 0$ for every i such that $g_i(x^*) < 0$ and hence, by definition of λ_i^k , we have

$$\lambda_i^k = \min\{g_i(x_k), 0\} = 0, \quad \text{for every } i \text{ such that } g_i(x^*) < 0,$$

so that, as λ_i^k converge a λ_i^* and $g_i(x^*) \leq 0$, we can write

$$\lambda_i^* g_i(x^*) = 0, \quad \text{for all } i = 1, \dots, m. \quad (5.12)$$

By (5.8), (5.10), (5.11) and (5.12) we obtain (5.2) and hence we have proved part (a) of our thesis, which establishes the FJ conditions.

Assertion (b) follows from the fact that $\lambda_i^* > 0$, implies $\lambda_i^k > 0$ for sufficiently large values of k and this implies in turn, by definition of λ_i^k , that $g_i(x_k) > 0$ for large values of k . Similarly, if $\mu_i^* \neq 0$, for large k the scalar μ_i^k has the same sign of μ_i^* and hence, by definition of μ_i^k , it has the same sign of $h_i(x_k)$. This implies that $\mu_i^* h_i(x_k) > 0$. As we have a finite number of constraints, in every neighborhood of x^* we can find, for sufficiently large k a point x_k such that

$$\lambda_i^* g_i(x_k) > 0 \quad \text{for all } i \text{ such that } \lambda_i^* > 0$$

$$\mu_i^* h_i(x_k) > 0 \quad \text{for all } i \text{ such that } \mu_i^* \neq 0.$$

This completes the proof. □

Letting

$$\nabla g = (\nabla g_1 \dots \nabla g_m)$$

$$\nabla h = (\nabla h_1 \dots \nabla h_p),$$

we can rewrite FJ conditions in matrix notation

Necessary Fritz John Conditions

$$\begin{aligned}\nabla_x \hat{L}(x^*, \lambda_0^*, \lambda^*, \mu^*) &\equiv \lambda_0^* \nabla f(x^*) + \nabla g(x^*) \lambda^* + \nabla h(x^*) \mu^* = 0 \\ (\lambda^*)^T g(x^*) &= 0, \\ (\lambda_0^*, \lambda^*) &\geq 0, \quad (\lambda_0^*, \lambda^*, \mu^*) \neq 0, \\ g(x^*) &\leq 0, \quad h(x^*) = 0.\end{aligned}\tag{5.13}$$

5.3 Constraint Qualifications and KKT Conditions

We have seen that in Fritz John conditions the gradient of the objective function is weighted with a scalar multiplier $\lambda_0 \geq 0$. If $\lambda_0 = 0$ then the objective function disappears from the optimality conditions, which is obviously unsatisfactory. Thus, it is of interest to establish under what conditions we can assume $\lambda_0^* > 0$. The conditions to be imposed on the constraints are known as *constraint qualifications*. Here we consider only some cases of special interest and we refer to the literature for a more comprehensive study. Preliminarily we consider some simple extensions, introduced in [179], of the usual notions of convexity and concavity, which can be useful for characterizing the local behavior of the constraints in a neighborhood of a given point $\bar{x} \in R^n$. With reference to continuously differentiable functions, we give the following definitions.

Definition 5.1 (Convexity and Concavity at a Point) Let $\bar{x} \in R^n$ a given point and let g_i a continuously differentiable function in a neighborhood $B(\bar{x}; \rho)$ of \bar{x} . We say that:

(i) g_i is *convex at the point \bar{x}* if

$$g_i(x) \geq g_i(\bar{x}) + \nabla g_i(\bar{x})^T (x - \bar{x}) \quad \text{for all } x \in B(\bar{x}; \rho);$$

(ii) g_i is *strictly convex at the point \bar{x}* if

$$g_i(x) > g_i(\bar{x}) + \nabla g_i(\bar{x})^T (x - \bar{x}) \quad \text{for all } x \in B(\bar{x}; \rho) \text{ with } x \neq \bar{x};$$

(continued)

Definition 5.1 (continued)(iii) g_i is *concave at the point \bar{x}* if

$$g_i(x) \leq g_i(\bar{x}) + \nabla g_i(\bar{x})^T(x - \bar{x}) \quad \text{for all } x \in B(\bar{x}; \rho);$$

(iv) g_i is *strictly concave at the point \bar{x}* if

$$g_i(x) < g_i(\bar{x}) + \nabla g_i(\bar{x})^T(x - \bar{x}) \quad \text{for all } x \in B(\bar{x}; \rho) \text{ with } x \neq \bar{x}.$$

We note that in the preceding definitions we consider only the pairs of points in $B(\bar{x}; \rho)$ such that one element is \bar{x} . It is easily seen that the preceding conditions are satisfied if g_i is (strictly) convex or concave in the neighborhood considered. We note also that if h_i is linear at x^* , then h_i is both convex and concave at x^* .

We consider now some constraint qualifications where we assume that $x^* \in S$ is a point satisfying the FJ conditions and we denote by $I_0(x^*)$ the index set of the inequality constraints that are active at x^* , that is:

$$I_0(x^*) = \{i : g_i(x^*) = 0\}.$$

(C1) Linear independence of active constraints at x^*

The set

$$\{\nabla h_i(x^*), \quad i = 1, \dots, p \quad \nabla g_i(x^*), \quad i \in I_0(x^*)\}.$$

is linearly independent.

□

(C2) Concavity at x^* of the constraints active at x^*

There exists a neighborhood $B(x^, \rho)$ of x^* such that, for all $x \in B(x^*, \rho)$ we have*

$$h_i(x) = h_i(x^*) + \nabla h_i(x^*)^T(x - x^*), \quad i = 1, \dots, p \quad (5.14)$$

$$g_i(x) \leq g_i(x^*) + \nabla g_i(x^*)^T(x - x^*), \quad i \in I_0(x^*). \quad (5.15)$$

□

An important special case of this condition is when all constraints are linear.

(C3) Mangasarian-Fromovitz conditions

We say that the Mangasarian-Fromovitz (MF) condition is satisfied if

- (i) *the gradient of the equality constraints are linearly independent;*

(ii) there exists $d \in R^n$ such that

$$\nabla g_i(x^*)^T d < 0, i \in I_0(x^*), \quad \nabla h_i(x^*)^T d = 0, i = 1, \dots, p.$$

□

Another interesting case is when the feasible set is defined only by convex inequality constraints and the following condition holds.

(C4) Slater's condition

Suppose that the functions g_i are convex and continuously differentiable on an open convex set containing the feasible set $S = \{x \in R^n : g(x) \leq 0\}$. We say that Slater's condition holds on S if there exists $\hat{x} \in S$ such that $g(\hat{x}) < 0$.

□

Now we show that, under the qualification conditions stated above, we can assume $\lambda_0^* > 0$ in the FJ conditions. In this case, we can divide the first equation of (5.2) by $\lambda_0^* > 0$ and then redefine the remaining multipliers, thus obtaining the Karush-Kuhn-Tucker conditions (KKT) where we have $\lambda_0^* = 1$.

Proposition 5.2 (Karush-Kuhn-Tucker Necessary Optimality Conditions) Let $x^* \in S$ be a local minimum point and suppose that the functions f, g, h are continuously differentiable in an open neighborhood of x^* .

Suppose further that at x^* one of the following constraints qualification holds.

- (C1) Linear independence of the constraints active at x^* .
- (C2) Concavity at x^* of the constraints active at x^* .
- (C3) Mangasarian-Fromovitz conditions.
- (C4) Slater's condition.

Then there exist $\lambda^* \in R^m$ and $\mu^* \in R^p$ such that the following KKT conditions hold.

$$\begin{aligned} \nabla f(x^*) + \nabla g(x^*)\lambda + \nabla h(x^*)\mu^* &= 0 \\ g(x^*) &\leq 0, \quad h(x^*) = 0 \\ \lambda^{*T} g(x^*) &= 0 \\ \lambda^* &\geq 0. \end{aligned} \tag{5.16}$$

Proof Under the assumptions stated we know that the FJ conditions are satisfied. As noted before, if $\lambda_0^* > 0$ then the KKT conditions hold for an appropriate definition of the multipliers and the assertion is true. Thus, reasoning by contradiction, we can suppose that $\lambda_0^* = 0$.

Consider first case (C1). Then, by FJ conditions, as $\lambda_i^* = 0$ for $i \notin I_0(x^*)$, we should find multipliers $(\lambda^*, \mu^*) \neq 0$ such that

$$\sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla h_i(x^*) = 0,$$

But the assumption (C1) would imply $\lambda^* = 0, \mu^* = 0$, which yields a contradiction.

In case (C2), from (5.14) and (5.15) by multiplying each constraint for the corresponding FJ multiplier and summing all equalities and inequalities we get

$$\begin{aligned} \sum_{i=1}^p \mu_i^* h_i(x) + \sum_{i \in I_0(x^*)} \lambda_i^* g_i(x) &\leq \sum_{i=1}^p \mu_i^* h_i(x^*) + \sum_{i \in I_0(x^*)} \lambda_i^* g_i(x^*) \\ &+ \left(\sum_{i=1}^p \mu_i^* \nabla h_i(x^*) + \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*) \right)^T (x - x^*) \end{aligned} \quad (5.17)$$

Then, recalling the FJ conditions, the assumption $\lambda_0^* = 0$ implies that the right hand side expression in (5.17) is zero and hence we have, for all $x \in B(x^*, \rho)$:

$$\sum_{i=1}^p \mu_i^* h_i(x) + \sum_{i \in I_0(x^*)} \lambda_i^* g_i(x) \leq 0. \quad (5.18)$$

However, if there exists at least a non zero FJ multiplier, by (b) of Proposition 5.1 we can find $x \in B(x^*, \rho)$ such that the sum in the left hand side is positive and hence we get a contradiction.

Suppose now that condition (C3) holds and that $\lambda_0^* = 0$. As the FJ multipliers cannot be all zero, there are only two possibilities:

- (a) $\lambda^* = 0, \mu^* \neq 0$;
- (b) there exists at least one $i \in I_0(x^*)$ such that $\lambda_i^* > 0$.

In case (a), from FJ conditions it follows that $\nabla h(x^*) \mu^* = 0$ with $\mu^* \neq 0$; on the other hand assumption (i) of MF condition would imply $\mu^* = 0$ and this yields a contradiction. In case (b), let d the vector considered in the MF condition. Then, recalling the FJ conditions we can write:

$$\nabla_x L(x^*, \lambda_0^*, \lambda^*, \mu^*)^T d = \sum_{i=1}^p \mu_i^* \nabla h_i(x^*)^T d + \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*)^T d = 0. \quad (5.19)$$

But this contradicts (ii) of MF condition, which implies

$$\sum_{i=1}^p \mu_i^* \nabla h_i(x^*)^T d + \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*)^T d = \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*)^T d < 0,$$

as in case (b) there must exist at least one $i \in I_0(x^*)$ such that $\lambda_i^* > 0$.

Finally we can easily verify that (C4) implies (C3). In fact, if Slater's condition holds at x^* , then for all $i \in I_0(x^*)$, by the convexity of g_i and the assumption $g_i(\hat{x}) < 0$, we have:

$$0 > g_i(\hat{x}) \geq g_i(x^*) + \nabla g_i(x^*)^T (\hat{x} - x^*) = \nabla g_i(x^*)^T (\hat{x} - x^*).$$

It follows that the MF condition is satisfied by choosing $d = \hat{x} - x^*$, which implies $\nabla g_i(x^*)^T d < 0$ for all $i \in I_0(x^*)$. Thus the assertion follows from case (C3). \square

If we define the Lagrangian function for Problem (5.1), by letting

$L(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x)$, we have:

$$\nabla_x L(x, \lambda, \mu) = \nabla f(x) + \nabla g(x)\lambda + \nabla h(x)\mu,$$

and the KKT conditions can be written in the following form.

Karush-Kuhn-Tucker Necessary Conditions

$$\begin{aligned} \nabla_x L(x^*, \lambda^*, \mu^*) &= 0 \\ g(x^*) &\leq 0, \quad h(x^*) = 0 \\ \lambda^{*T} g(x^*) &= 0 \\ \lambda^* &\geq 0. \end{aligned} \tag{5.20}$$

Example 5.1 Consider the problem

$$\begin{aligned} \min \quad & 2x_2 - x_1^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 \leq 1 \\ & x_1 \geq 0, \quad x_2 \geq 0. \end{aligned}$$

We first observe that the feasible set is non empty and compact and hence the minimization problem must admit a global solution. We can also note that the feasible set is convex and that the Slater condition is satisfied (take, for instance, $\hat{x} = (1/2, 1/2)$). Then we know that KKT conditions must hold at a minimizer.

The Lagrangian function can be put in the form

$$L(x, \lambda) = 2x_2 - x_1^2 - \lambda_1 x_1 - \lambda_2 x_2 + \lambda_3 (x_1^2 + x_2^2 - 1).$$

Then the KKT conditions become:

$$\begin{aligned}
 (1) \quad & -2x_1 - \lambda_1 + 2\lambda_3 x_1 = 0 \\
 (2) \quad & 2 - \lambda_2 + 2\lambda_3 x_2 = 0, \\
 (3) \quad & x_1^2 + x_2^2 \leq 1 \\
 (4) \quad & x_1 \geq 0, \quad x_2 \geq 0 \\
 (5) \quad & \lambda_1 x_1 = 0, \lambda_2 x_2 = 0, \lambda_3(x_1^2 + x_2^2 - 1) = 0, \\
 (6) \quad & \lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0
 \end{aligned} \tag{5.21}$$

If we can solve this system, we would obtain a *critical point*, that is a candidate to be a minimizer of the problem.

As the problem is quite simple, we can attempt to find an analytical solution. A typical approach can be that of starting from the complementarity conditions and analyzing the different alternatives.

Consider first the multiplier $\lambda_1 \geq 0$. If $\lambda_1 > 0$ then, by (5), we have $x_1 = 0$, but then, by (1), we would obtain $\lambda_1 = 0$, which is a contradiction. Thus we can assume $\lambda_1 = 0$ and (1) will be replaced by

$$(1') \quad x_1(\lambda_3 - 1) = 0.$$

Next consider $\lambda_2 \geq 0$. If $\lambda_2 = 0$, from (2) we would obtain $\lambda_3 x_2 = -1$, which would contradict the nonnegativity constraints. Thus we must have $\lambda_2 > 0$, which in turn implies, by (5), $x_2 = 0$. Finally consider the multiplier $\lambda_3 \geq 0$.

If $\lambda_3 = 0$, then from (1') above we obtain $x_1 = 0$ and hence a solution to the KKT system is

$$\hat{x} = (0 \ 0), \quad \hat{\lambda} = (0, 2, 0).$$

However, we can assume also $\lambda_3 > 0$. In this case, we have from (5) that $x_1 = 1$ and from (1') that $\lambda_3 = 1$. Thus a solution of the KKT system would be also

$$x^* = (1 \ 0), \quad \lambda^* = (0, 2, 1),$$

and x^* is clearly the global minimizer. \square

In the special case of problems with only equality constraint we can construct the Lagrangian function $L(x, \mu) = f(x) + \mu^T h(x)$, and from the preceding result we get the well-known Lagrange multiplier rule, which we state in the following form.

Proposition 5.3 (Lagrange Multipliers) *Let $x^* \in R^n$ be a local minimum point of the problem*

$$\min f(x), \quad \text{subject to} \quad h(x) = 0,$$

and suppose that f, h are continuously differentiable in a neighborhood of x^ . Suppose further that one of the following constraint qualifications holds.*

- (C1) Linear independence of the gradients $\{\nabla h_i(x^*), i = 1, \dots, p\}$.
- (C2) Linearity of the constraints at x^* .

Then there exists $\mu^ \in R^p$ such that*

$$\nabla f(x^*) + \nabla h(x^*)\mu^* = 0 \quad h(x^*) = 0.$$

□

When the Lagrange multiplier rule is valid, then the search for a critical point satisfying the necessary optimality conditions consists in solving the system of $n + p$ equations in the $n + p$ variables:

$$\nabla_x L(x, \mu) = 0 \quad h(x) = 0.$$

Example 5.2 As an example of application of the Lagrange multiplier rule, let us consider the derivation of a well known result in linear Algebra.

In particular, we establish that any vector $b \in R^m$ can be represented as

$$b = b_1 + b_2, \quad b_1 \in \mathcal{R}(A), \quad b_2 \in \mathcal{N}(A^T), \quad (5.22)$$

where A is any $m \times n$ real matrix, $\mathcal{R}(A)$ and $\mathcal{N}(A^T)$ are, respectively the range space of A and the null space of A^T , that is. $\mathcal{R}(A) = \{y \in R^m : y = Ax, x \in R^n\}$, and $\mathcal{N}(A^T) = \{y \in R^m : A^T y = 0\}$.

Consider the problem

$$\min \frac{1}{2} \|b - y\|^2$$

$$A^T y = 0.$$

This problem admits an optimal solution, as the feasible set is non empty and closed and the objective function is coercive on it. Since the constraints are linear we

know that an optimal solution y^* must satisfy the Lagrange multiplier rule. The Lagrangian function is

$$L(y, \mu) = \frac{1}{2} \|b - y\|^2 + \mu^T A^T y = \frac{1}{2} \|b - y\|^2 + (A\mu)^T y.$$

Therefore, there must exist a multiplier $\mu^* \in R^n$ such that

$$-(b - y^*) + A\mu^* = 0, \quad A^T y^* = 0.$$

It follows that $b = y^* + A\mu^*$, with $A^T y^* = 0$. This gives the representation of b as sum of y^* in the null space of A^T and a vector $A\mu^*$ in the range space of A .

We can also observe that μ^* is a solution of the *normal equations*. In fact, by premultiplying both members of the last equation by A^T and recalling that $A^T y^* = 0$, we obtain $A^T A\mu^* = A^T b$.

5.4 Sufficient Conditions in the Convex Case

We can show that the KKT conditions become *sufficient conditions of global optimality* when f and all g_i are convex functions and the equality constraints are linear.

Proposition 5.4 (Sufficiency of KKT Conditions in the Convex Case)

Assume that the functions f , g_i , for $i = 1, \dots, m$ are convex and that the equality constraints h_i are defined by affine functions, that is $h(x) \equiv Ax - b$, for some $A p \times n$ and $b \in R^p$. Assume also that all the problem functions are continuously differentiable on an open set containing the feasible set.

Then, if there exist multipliers λ^* and μ^* such that the KKT conditions are satisfied at a point $x^* \in R^n$, that is

$$\begin{aligned} \nabla f(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* &= 0 \\ g(x^*) &\leq 0, \quad h(x^*) = 0 \\ \lambda^{*T} g(x^*) &= 0 \\ \lambda^* &\geq 0, \end{aligned}$$

the point x^* is a global minimum point of the constrained problem; moreover, if f is strictly convex, the point x^* is the unique global solution.

Proof If x is a feasible point, as $g(x) \leq 0$, $h(x) = 0$ and $\lambda^* \geq 0$, we can write

$$f(x) \geq f(x) + \lambda^{*T} g(x) + \mu^{*T} h(x).$$

Moreover, by the linearity of the equality constraints, we have

$$h(x) = h(x^*) + \nabla h(x^*)^T (x - x^*).$$

The convexity of all functions g_i implies that

$$g_i(x) \geq g_i(x^*) + \nabla g_i(x^*)^T (x - x^*), \quad i = 1, \dots, m$$

and hence, as $\lambda^* \geq 0$, we obtain

$$\lambda^{*T} g(x) \geq \lambda^{*T} g(x^*) + \lambda^{*T} \nabla g(x^*)^T (x - x^*).$$

Similarly, the convexity of f is implies

$$f(x) \geq f(x^*) + \nabla f(x^*)^T (x - x^*).$$

Then we can write

$$\begin{aligned} f(x) &\geq f(x) + \lambda^{*T} g(x) + \mu^{*T} h(x) \\ &\geq f(x^*) + \nabla f(x^*)^T (x - x^*) + \lambda^{*T} g(x^*) + \lambda^{*T} \nabla g(x^*)^T (x - x^*) \\ &\quad + \mu^{*T} h(x^*) + \mu^{*T} \nabla h(x^*)^T (x - x^*) \\ &\geq f(x^*) + (\nabla f(x^*) + \nabla g(x^*) \lambda^* + \nabla h(x^*) \mu^*)^T (x - x^*) \\ &= f(x^*). \end{aligned}$$

This shows that $f(x) \geq f(x^*)$ for all feasible x .

If f is strictly convex, we have $f(x) > f(x^*)$ for $x \neq x^*$, and this shows that the minimum point is unique. \square

5.5 Second Order Optimality Conditions

Assuming that the problem functions are twice continuously differentiable, we can obtain second order optimality conditions under suitable constraint qualifications. Preliminarily we state the following well known result, which we establish using the Lagrange multiplier rule.

Lemma 5.1 Let M be a real $n \times q$ matrix, with $q \leq n$ and suppose that M has rank q . Then, given $y \in R^n$, the projection $\pi(y)$ of y in the null space $\mathcal{N}(M^T)$ of M^T , is given by

$$\pi(y) = y - M(M^T M)^{-1} M^T y. \quad (5.23)$$

Proof The projection on $\mathcal{N}(M^T)$ is the solution of the problem

$$\begin{aligned} \min \frac{1}{2} \|z - y\|^2 \\ M^T z = 0, \end{aligned}$$

As the constraints are linear, we can use the Lagrange multiplier rule and we can find μ^* such that

$$\pi(y) - y + M\mu^* = 0.$$

Pre-multiplying by M^T and then solving with respect to μ^* , as $\pi(y)$ is feasible, so that $M^T \pi(y) = 0$, we obtain $\mu^* = (M^T M)^{-1} M^T y$ and hence we get (5.23). \square

Then we can establish the following result. See, e.g. [92].

Proposition 5.5 (Second Order Necessary Conditions) Let x^* be a local minimum point of Problem (5.1) and suppose that the functions f, g, h are twice continuously differentiable on an open neighborhood of x^* .

Suppose that the qualification condition (C1) holds at x^* . Then there exist multipliers $\lambda^* \in R^m$, $\mu^* \in R^p$ such that the KKT conditions are satisfied and we have

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d \geq 0,$$

for all $d \in R^n$ such that

$$\nabla h_i(x^*)^T d = 0, \quad i = 1, \dots, p \quad \nabla g_i(x^*)^T d = 0, \quad i \in I_0(x^*).$$

Proof Under the assumptions stated, as the constraint qualification (C1) is satisfied, by Proposition 5.2 we have that the KKT conditions hold. Now define the same auxiliary function introduced in the proof of Proposition 5.1, that is:

$$F_k(x) = f(x) + \frac{1}{4} \|x - x^*\|^4 + \frac{k}{3} \sum_{i=1}^m (g_i^+(x))^3 + \frac{k}{2} \sum_{i=1}^p h_i^2(x),$$

where $g_i^+(x) = \max\{g_i(x), 0\}$. Under the assumptions stated here it can be easily verified that this function (and hence also F_k) is twice continuously differentiable in a sufficiently small neighborhood of x^* and that we have

$$(1/3)\nabla^2(g_i^+(x))^3 = (g_i^+(x))^2 \nabla^2 g_i(x) + 2g_i^+(x) \nabla g_i(x) \nabla g_i(x)^T.$$

Reasoning exactly as in the proof of Proposition 5.1 (possibly in a smaller neighborhood, if needed) let us construct the sequence of points $\{x_k\}$ converging to x^* defined there and the bounded sequences of FJ multiplier estimates converging to multipliers such that the FJ conditions hold. Since the constraint qualification condition (C1) is satisfied we know also that the multiplier λ_0^* associated to f must be positive.

As the problem functions are twice continuously differentiable, we can choose k sufficiently large to have that the second order necessary condition for an unconstrained minimum holds at x_k , so that, for each k we have, for all $y \in R^n$:

$$\begin{aligned} 0 \leq y^T \nabla^2 F_k(x_k) y &\equiv y^T \left(\nabla^2 f(x_k) + \sum_{i=1}^p k h_i(x_k) \nabla^2 h_i(x_k) \right. \\ &\quad \left. + \sum_{i=1}^m k (g_i^+(x_k))^2 \nabla^2 g_i(x_k) \right) y \\ &\quad + \sum_{i=1}^p k (\nabla h_i(x_k)^T y)^2 + \sum_{i=1}^m 2k g_i^+(x_k) (\nabla g_i(x_k)^T y)^2 \\ &\quad + \|x_k - x^*\|^2 \|y\|^2 + 2[(x_k - x^*)^T y]^2. \end{aligned} \tag{5.24}$$

Now, define the matrix M with columns

$$\nabla h_i(x^*), i = 1, \dots, p \quad \nabla g_i(x^*), i \in I_0(x^*).$$

By assumption, this matrix has full rank and the matrix $M^T M$ is non singular. Then, we can assume that k is large enough to have that also the matrix M_k with columns

$$\nabla h_i(x_k), i = 1, \dots, p \quad \nabla g_i(x_k), i \in I_0(x^*)$$

has full rank. Let d be a given vector in R^n such that $M^T d = 0$, and, for each sufficiently large k , compute the projection d_k of d on the null space of M_k^T . Recalling (5.23), we have

$$d_k = d - M_k(M_k^T M_k)^{-1} M_k^T d, \quad (5.25)$$

so that $M_k^T d_k = 0$, that is $\nabla h_i(x_k)^T d_k = 0, i = 1, \dots, p$ $\nabla g_i(x_k)^T d_k = 0, i \in I_0(x^*)$.

Then, if take $y = d_k$ in (5.24), and we choose k sufficiently large to have $g_i^+(x_k) = 0, i \notin I_0(x^*)$, we can write

$$\begin{aligned} 0 \leq d_k^T \nabla^2 F_k(x_k) d_k &= d_k^T \left(\nabla^2 f(x_k) + \sum_{i=1}^p k h_i(x_k) \nabla^2 h_i(x_k) \right. \\ &\quad \left. + \sum_{i=1}^m k(g_i^+(x))^2 \nabla^2 g_i(x_k) \right) d_k \\ &\quad + \|x_k - x^*\|^2 \|y\|^2 + 2[(x_k - x^*)^T y]^2. \end{aligned} \quad (5.26)$$

By the continuity assumptions, we have that for $k \rightarrow \infty$, as x_k converges to x^* , we have that $M_k \rightarrow M$ and hence that $M_k d \rightarrow 0$. Then from (5.25) it follows that $d_k \rightarrow d$. Taking this into account, and reasoning as in the proof of Proposition 5.1, from (5.26), taking limits, and dividing the FJ multipliers by the multiplier $\lambda_0^* > 0$ associated to f , we obtain:

$$d^T \left(\nabla^2 f(x^*) + \sum_{i=1}^p \mu_i^* \nabla^2 h_i(x^*) + \sum_{i=1}^m \lambda_i^* \nabla^2 g_i(x^*) \right) d \geq 0, \quad (5.27)$$

where the multipliers are now the KKT multipliers. As this is true for every d such that $M^T d = 0$, the proof is complete. \square

A second order sufficient condition can be obtained by imposing appropriate conditions on the Hessian matrix of the Lagrangian function. One of the best known conditions, is reported in the next proposition. Given a KKT point (x^*, λ^*, μ^*) , we denote by $I_0^+(x^*)$ the index set of *strongly active constraints*, that is

$$I_0^+(x^*) = \{i \in I_0(x^*) : \lambda_i^* > 0\}. \quad (5.28)$$

We suppose, for brevity, that if some index set is empty the terms where this index appears must be omitted. Then we can state the following condition, established in [92].

Proposition 5.6 (Second Order Sufficiency Conditions) *Let x^* be a given point in \mathbb{R}^n and suppose that the functions f, g, h are twice continuously differentiable on an open neighborhood of x^* .*

Suppose there exist multipliers $\lambda^ \in \mathbb{R}^m$, $\mu^* \in \mathbb{R}^p$ such that the KKT conditions are satisfied at x^* . Assume further that*

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0,$$

for all $d \neq 0 \in \mathbb{R}^n$ such that

$$\begin{aligned}\nabla h_i(x^*)^T d &= 0, \quad i = 1, \dots, p \\ \nabla g_i(x^*)^T d &\leq 0, \quad i \in I_0(x^*) \\ \nabla g_i(x^*)^T d &= 0, \quad i \in I_0^+(x^*).\end{aligned}$$

Then x^ is a strict local minimizer of f subject to the constraints*

$$h(x) = 0, g(x) \leq 0.$$

Proof Suppose that the assertion is false. Then we can construct a sequence of feasible points x_k converging to x^* such that

$$f(x_k) \leq f(x^*). \quad (5.29)$$

Let λ^*, μ^* be the KKT multipliers associated to x^* . By the differentiability assumptions, defining $d_k = x_k - x^*$, we can write

$$\begin{aligned}L(x_k, \lambda^*, \mu^*) &= L(x^*, \lambda^*, \mu^*) + \nabla_x L(x^*, \lambda^*, \mu^*)^T d_k \\ &\quad + \frac{1}{2} d_k^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d_k + \beta_L(d_k),\end{aligned}$$

where

$$\lim_{k \rightarrow \infty} \frac{\beta_L(d_k)}{\|d_k\|^2} = 0.$$

Recalling the KKT conditions and (5.29) we can write

$$0 \geq f(x_k) + \lambda^{*T} g(x_k) - f(x^*) = \frac{1}{2} d_k^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d_k + \beta_L(d_k),$$

Therefore, recalling assumption (5.29) and dividing by $\|d_k\|^2$ we have

$$\frac{1}{2\|d_k\|^2} d_k^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d_k + \frac{\beta_L(d_k)}{\|d_k\|^2} \leq 0.$$

Then, taking limits for $k \rightarrow \infty$ we have that the vector $d_k/\|d_k\|$ converges to some d with $\|d\| = 1$ and we obtain:

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d \leq 0. \quad (5.30)$$

Now, the differentiability assumptions and the feasibility constraints imply that

$$f(x_k) = f(x^*) + \nabla f(x^*)^T d_k + \gamma_0(d_k), \quad (5.31)$$

$$0 \geq g_i(x_k) = g_i(x^*) + \nabla g_i(x^*)^T d_k + \gamma_i(d_k), i = 1, \dots, m \quad (5.32)$$

$$0 = h_i(x_k) = h_i(x^*) + \nabla h_i(x^*)^T d_k + \delta_i(d_k), i = 1, \dots, p \quad (5.33)$$

where

$$\lim_{k \rightarrow \infty} \gamma_i(d_k)/\|d_k\| = 0, i = 0, 1, \dots, m. \quad \lim_{k \rightarrow \infty} \delta_i(d_k)/\|d_k\| = 0, i = 1, \dots, p.$$

Taking into account (5.29), from (5.31) we obtain

$$\nabla f(x^*)^T d_k + \gamma_0(d_k) \leq 0, \quad (5.34)$$

Now, from the preceding inequality and from (5.32) and (5.33), dividing by $\|d_k\|$, taking limits and recalling that $d_k/\|d_k\| \rightarrow d$, we obtain

$$\nabla f(x^*)^T d \leq 0, \quad \nabla g_i(x^*)^T d \leq 0, \quad i \in I_0(x^*) \quad \nabla h_i(x^*)^T d = 0, \quad i = 1, \dots, p. \quad (5.35)$$

If $I_0^+(x^*)$ is empty from (5.30) and (5.35) we get a contradiction with the hypotheses of the proposition. If $I_0^+(x^*)$ is non empty, by the KKT conditions we have

$$\nabla f(x^*)^T d + \sum_{i \in I_0^+(x^*)} \lambda_i^* \nabla g_i(x^*)^T d + \sum_{i=1}^p \mu_i^* \nabla h_i(x^*)^T d = 0, \quad (5.36)$$

whence we get

$$\nabla f(x^*)^T d + \sum_{i \in I_0^+(x^*)} \lambda_i^* \nabla g_i(x^*)^T d = 0. \quad (5.37)$$

As $\lambda_i^* > 0$ for all $i \in I_0^+(x^*)$, from the preceding equation and (5.35) we must have necessarily that

$$\nabla g_i(x^*)^T d = 0, \quad i \in I_0^+(x^*). \quad (5.38)$$

It can be concluded that if the assertion is false, then by (5.30), (5.35), and (5.38) we get again a contradiction with the hypotheses of the proposition. \square

It is also possible to state “weaker” or “stronger” sufficient conditions (in the sense that they impose less or more restrictive conditions on the KKT point). A first possibility is that of obtaining what sometimes is called the *weak second order sufficient condition* that can be stated in the following form.

Proposition 5.7 (Weak Second Order Sufficiency Conditions) *Let x^* be a given point in R^n and suppose that the functions f, g, h are twice continuously differentiable on an open neighborhood of x^* .*

Suppose also that the set \mathcal{M}^ of KKT multipliers (λ^*, μ^*) such that the KKT conditions hold at x^* is non empty. Assume further that, for every $d \neq 0$ such that $\nabla f(x^*)^T d = 0$ and*

$$\begin{aligned} \nabla h_i(x^*)^T d &= 0, \quad i = 1, \dots, p \\ \nabla g_i(x^*)^T d &\leq 0, \quad i \in I_0(x^*). \end{aligned}$$

we have $\sup_{(\lambda^, \mu^*) \in \mathcal{M}^*} d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0$.*

Then x^ is a strict local minimizer of f subject to the constraints $h(x) = 0, g(x) \leq 0$. \square*

A proof of this result can be found, for instance, in [231] and can be obtained by suitably modifying the order of the arguments used in the proof of Proposition 5.6. We note that the main difference with respect to Proposition 5.6 is that in the weak condition the KKT multipliers (if not uniquely defined) can be chosen in dependence of the direction d .

Stronger conditions, which can be checked more easily, can be obtained from Proposition 5.6 by enlarging the region where $d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0$. An immediate consequence is the following condition, which is called by some authors (see e.g. [231]) the *semi-strong second order sufficient condition* and by others (see, e.g. [170]) as the *strong second order sufficient condition*.

Proposition 5.8 ((Semi-) Strong Second Order Sufficiency Conditions)

Let x^* be a given point in R^n and suppose that the functions f, g, h are twice continuously differentiable on an open neighborhood of x^* .

Suppose there exist multipliers $\lambda^* \in R^m$, $\mu^* \in R^p$ such that the KKT conditions are satisfied at x^* . Assume further that

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0,$$

for all $d \neq 0 \in R^n$ such that

$$\begin{aligned} \nabla h_i(x^*)^T d &= 0, \quad i = 1, \dots, p \\ \nabla g_i(x^*)^T d &= 0, \quad i \in I_0^+(x^*). \end{aligned}$$

Then x^* is a strict local minimizer of f subject to the constraints $h(x) = 0, g(x) \leq 0$. \square

A stronger condition is satisfied if we assume that strict complementarity holds at x^* .

Proposition 5.9 (Second Order Sufficiency Conditions with Strict Complementarity)

Let x^* be a given point in R^n and suppose that the functions

f, g, h are twice continuously differentiable on an open neighborhood of x^* .

Suppose there exist multipliers $\lambda^* \in R^m$, $\mu^* \in R^p$ such that the KKT conditions are satisfied at x^* .

Assume further that strict complementarity holds at x^* , that is

$$\lambda_i^* > 0 \quad \text{for all } i \in I_0(x^*)$$

and that

$$d^T \nabla_x^2 L(x^*, \lambda^*, \mu^*) d > 0,$$

for all $d \neq 0 \in R^n$ such that

$$\begin{aligned} \nabla h_i(x^*)^T d &= 0, \quad i = 1, \dots, p \\ \nabla g_i(x^*)^T d &= 0, \quad i \in I_0(x^*). \end{aligned}$$

(continued)

Proposition 5.9 (continued)

Then x^ is a strict local minimizer of f subject to the constraints*

$$h(x) = 0, g(x) \leq 0.$$

□

5.6 Linearly Constrained Problems

In Chap. 4 we have already considered optimization problems with a convex feasible set defined by linear constraints and we have shown that the optimality conditions can be expressed as *non existence* of feasible descent directions at a minimizer. Optimality conditions expressed as a system of linear inequalities can be obtained by employing the theorems of the alternative reviewed in the appendix to Chap. 7. Here we reobtain these conditions as special case of the KKT conditions. Without loss of generality, we can refer to an inequality constrained problems of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & Ax \geq b, \end{aligned} \tag{5.39}$$

where f is a continuously differentiable function and A is a real $m \times n$ matrix.

We already know that the constraint qualification condition (C2) is satisfied in problem (5.39) and hence that the optimality conditions can be stated as a special case of Proposition 5.2.

Proposition 5.10 (KKT Conditions for Linearly Constrained Problems)

Let x^ be a local minimum point of Problem (5.39). Then there exists $\lambda^* \in R^m$ such that the following conditions are satisfied, where the Lagrangian function is defined as*

$$L(x, \lambda) = f(x) + \lambda^T(b - Ax).$$

- (i) $Ax^* \geq b$;
- (ii) $\nabla_x L(x^*, \lambda^*) = \nabla f(x^*) - A^T \lambda^* = 0$;
- (iii) $\lambda^* \geq 0$,
- (iv) $\lambda^{*T}(b - Ax^*) = 0$.

When f is convex we can state the following condition, which follows from Propositions 5.10 and 5.4.

Proposition 5.11 (Necessary and Sufficient Global Optimality Conditions) Let $S = \{x : x \in R^n, Ax \geq b\}$, let $D \subseteq R^n$ an open convex set containing S , let $f : D \rightarrow R$ and assume that f is convex and continuously differentiable on D . Then a point x^* is a global minimum point of f on S if and only if there exists $\lambda^* \in R^m$ such that the KKT conditions stated in Proposition 5.10 are satisfied. Moreover, if f is strictly convex on D and the KKT conditions hold, then x^* is the unique global minimum point of f on S .

Some interesting special cases of linearly constrained problems are considered in the next subsections.

5.6.1 Problems with Non Negativity Constraints

Consider a problem of the form

$$\min f(x), \quad x \geq 0, \quad (5.40)$$

where $f : R^n \rightarrow R$ is a continuously differentiable function. We have

$$L(x, \lambda) = f(x) - \lambda^T x, \quad \nabla_x L(x, \lambda) = \nabla f(x) - \lambda$$

and hence, using the KKT conditions at x^* , we obtain

$$\nabla f(x^*) - \lambda^* = 0, \quad x^* \geq 0, \quad \lambda^* \geq 0, \quad \lambda^{*T} x^* = 0.$$

By considering the single components, we have

$$\lambda_j^* = \frac{\partial f(x^*)}{\partial x_j}, \quad j = 1, \dots, n$$

so that the KKT conditions are equivalent to

$$\frac{\partial f(x^*)}{\partial x_j} \geq 0 \quad \text{if } x_j^* = 0, \quad \frac{\partial f(x^*)}{\partial x_j} = 0 \quad \text{if } x_j^* > 0.$$

If f is convex, these conditions become necessary and sufficient conditions of global optimality.

5.6.2 Problems with Box Constraints

Consider the problem

$$\begin{aligned} \min f(x) \\ a \leq x \leq b, \end{aligned} \tag{5.41}$$

where $a, b \in R^n$ and we assume $b > a$. We can define the Lagrangian function

$$L(x, u, v) = f(x) + u^T(a - x) + v^T(x - b),$$

where $(u, v) \in R^n \times R^n$ are vectors of KKT multipliers. The KKT conditions at a point x^* such that $a \leq x^* \leq b$, become

$$\nabla f(x^*) - u^* + v^* = 0, \quad (a - x^*)^T u^* = 0, \quad (x^* - b)^T v^* = 0, \quad (u^*, v^*) \geq 0.$$

If we introduce the index sets

$$J_a = \{j : x_j^* = a_j\}, \quad J_b = \{j : x_j^* = b_j\}, \quad J_0 = \{j : a_j < x_j^* < b_j\}$$

we can express the optimality conditions in correspondence to each index set.

- (i) If $j \in J_a$ we have necessarily $x_j^* < b_j$ and hence by the complementarity conditions we have $v_j^* = 0$. It follows that

$$\frac{\partial f(x^*)}{\partial x_j} = u_j^* \geq 0.$$

- (ii) If $j \in J_b$ we have $x_j^* > a_j$ and $u_j^* = 0$, so that

$$\frac{\partial f(x^*)}{\partial x_j} = -v_j^* \leq 0.$$

- (iii) if $j \in J_0$ we have $u_j^* = 0$ e $v_j^* = 0$, and hence

$$\frac{\partial f(x^*)}{\partial x_j} = 0.$$

We can summarize the necessary optimality conditions (which are necessary and sufficient in the convex case):

$$\begin{aligned}\frac{\partial f(x^*)}{\partial x_j} &\geq 0 \quad \text{if } x_j^* = a_j, \\ \frac{\partial f(x^*)}{\partial x_j} &\leq 0 \quad \text{if } x_j^* = b_j, \\ \frac{\partial f(x^*)}{\partial x_j} &= 0 \quad \text{if } a_j < x_j^* < b_j, \\ a &\leq x^* \leq b.\end{aligned}$$

5.6.3 Problems with Simplex Constraints

We consider now the problems with simplex-type constraints, of the form

$$\begin{aligned}\min f(x) \\ e^T x = 1, \quad x \geq 0\end{aligned}\tag{5.42}$$

where $e = (1, 1 \dots, 1)^T \in R^n$. The Lagrangian function is given by:

$$L(x, \mu, \lambda) = f(x) + \mu(e^T x - 1) - \lambda^T x,$$

where $\mu \in R$ and $\lambda \in R^n$. From the KKT conditions we get

$$\begin{aligned}\nabla f(x^*) + \mu^* e - \lambda^* &= 0, \\ e^T x = 1, \quad x \geq 0 \\ x^{*T} \lambda^* &= 0, \quad \lambda^* \geq 0.\end{aligned}$$

We can write, in terms of the single components:

$$\frac{\partial f(x^*)}{\partial x_j} - \lambda_j^* = -\mu^*, \quad j = 1, \dots, n.\tag{5.43}$$

If $x_j^* > 0$ we have, by complementarity, $\lambda_j^* = 0$ and hence

$$\frac{\partial f(x^*)}{\partial x_j} = -\mu^*, \quad \text{for every } j \text{ such that } x_j^* > 0.$$

On the other hand, as $\lambda^* \geq 0$, from (5.43) it follows that

$$\frac{\partial f(x^*)}{\partial x_h} \geq -\mu^*, \quad h = 1, \dots, n.$$

Therefore, the optimality conditions consist in imposing that

$$x_j^* > 0 \quad \text{implies} \quad \frac{\partial f(x^*)}{\partial x_j} \leq \frac{\partial f(x^*)}{\partial x_h} \quad \text{for every } h = 1, \dots, n,$$

with the constraints $e^T x^* = 1$, $x^* \geq 0$. These conditions, as we know, become also necessary and sufficient conditions of global optimum when f is convex.

5.6.4 Quadratic Programming

An important class of convex programming problems is that of (convex) *quadratic programming* problems, of the form

$$\min \frac{1}{2} x^T Q x + c^T x, \quad (\text{QP})$$

$$Ax \geq b$$

where Q is a $n \times n$ symmetric, positive semidefinite matrix.

From Proposition 5.11, by the KKT conditions, noting that

$$\nabla_x L(x, \lambda) = Qx + c - A^T \lambda,$$

we have the following necessary and sufficient conditions of global optimum.

Proposition 5.12 (Optimality Conditions for QP) *Let Q be a symmetric, positive semidefinite matrix. Then a necessary and sufficient condition for x^* to be a global minimizer of problem (QP) is that there exists $\lambda^* \in R^m$ such that*

- (i) $\nabla_x L(x^*, \lambda^*) = Qx^* + c - A^T \lambda^* = 0$;
- (ii) $Ax^* \geq b$;
- (iii) $\lambda^{*T} (b - Ax^*) = 0$, $\lambda^* \geq 0$.

When Q is positive definite then x^* is the unique global minimizer of problem (QP). \square

We can note that, if Q is positive semidefinite, could not exist an optimal solution, even if the feasible set is non empty and, in this case, the KKT conditions cannot hold. When Q is positive definite the level sets of f are compact and hence, if the feasible set is non empty, there exists an optimal solution satisfying the KKT conditions.

A special case, where we can obtain analytically the optimal solution from the optimality conditions is the following equality constrained problem.

$$\min \frac{1}{2}x^T Qx + c^T x,$$

$$Ax = b,$$

where Q is positive definite and the matrix $A(m \times n)$ has rank $m < n$. From the Lagrange multiplier rule we obtain the conditions

$$Qx^* + c - A^T \mu^* = 0, \quad Ax^* = b.$$

Solving with respect to x^* , we have $x^* = -Q^{-1}(c - A^T \mu^*)$, and hence, by imposing the constraint, we get $A(-Q^{-1}(c - A^T \mu^*)) = b$, whence it follows:

$$AQ^{-1}A^T \mu^* = b + AQ^{-1}c.$$

As $AQ^{-1}A^T$ is non singular, we have

$$\mu^* = \left(AQ^{-1}A^T \right)^{-1} (b + AQ^{-1}c)$$

and we obtain the optimal solution

$$x^* = -Q^{-1}c + Q^{-1}A^T \left(AQ^{-1}A^T \right)^{-1} (b + AQ^{-1}c).$$

5.6.5 Linear Programming

From Proposition 5.12, assuming $Q = 0$, we obtain the optimality conditions for Linear Programming (LP). In particular, if we refer to LP problems in the form

$$\min c^T x, \quad (\text{LP})$$

$$Ax \geq b$$

and we let

$$L(x, \lambda) = c^T x + \lambda^T (b - Ax)$$

we have

$$\nabla_x L(x, \lambda) = c - A^T \lambda$$

and hence we get the following condition.

Proposition 5.13 (Optimality Conditions for (LP)) *A necessary and sufficient condition for x^* to be a global minimum point of Problem (LP) is that there exists $\lambda^* \in R^m$ such that:*

- (i) $\nabla_x L(x^*, \lambda^*) \equiv c - A^T \lambda^* = 0;$
- (ii) $Ax^* \geq b;$
- (iii) $\lambda^{*T} (b - Ax^*) = 0, \quad \lambda^* \geq 0.$

□

The optimality conditions can be reformulated, noting that the conditions stated in Propositions 5.13 imply, in particular:

$$\lambda^{*T} b = \lambda^{*T} Ax^* = (A^T \lambda^*)^T x^* = c^T x^*.$$

Thus we get an optimality condition expressed as a system of linear equations and inequalities in the pair (x, λ) .

Proposition 5.14 (Optimality Conditions for (LP)) *A necessary and sufficient condition for x^* to be a global minimum point of Problem (LP) is that there exists $\lambda^* \in R^m$ such that:*

- (i) $\nabla_x L(x^*, \lambda^*) \equiv c - A^T \lambda^* = 0;$
- (ii) $Ax^* \geq b;$
- (iii) $b^T \lambda^* = c^T x^*, \quad \lambda^* \geq 0.$

□

5.7 Exercises

Using the optimality conditions solve and discuss the following five problems.

5.1

$$\begin{aligned} & \max a^T x \\ & x^T Q x \leq 1, \end{aligned}$$

where Q is a positive definite symmetric matrix and $a \in R^n$ is a given vector.

5.2

$$\begin{aligned} & \min x^T A x \\ & \|x\|^2 = 1, \end{aligned}$$

where A is a real symmetric matrix.

5.3

$$\begin{aligned} & \max x_1 x_2 x_3 \\ & x_1 + x_2 + x_3 \leq 1 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \end{aligned}$$

5.4

$$\begin{aligned} & \min e^T x \\ & \|x\|^2 = 1, \end{aligned}$$

where $e = (1, \dots, 1)^T$.

5.5

$$\begin{aligned} & \min \|x\|^2 \\ & e^T x = 1, \end{aligned}$$

where $e = (1, \dots, 1)^T$.

5.6 Determine the Euclidean distance of a point $x_0 \in R^n$ from the hyperplane

$$a^T x = \beta,$$

by solving the problem

$$\begin{aligned} & \min \|x - x_0\|^2 \\ & a^T x = \beta. \end{aligned}$$

5.7 Determine an optimal solution of the problem

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

5.8 Consider the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & h_j(x) = 0 \quad j = 1, \dots, p, \end{aligned}$$

where $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$, $h : R^n \rightarrow R^p$ are continuously differentiable functions. Let x^* be a feasible point satisfying the KKT conditions. Show that, given any direction $d \in R^n$ such that

$$\begin{aligned} \nabla g_i(x^*)^T d \leq 0 & \quad \text{for every } i \in I_0(x^*) \\ \nabla h_j(x^*)^T d = 0 & \quad i = 1, \dots, p, \end{aligned}$$

we have

$$\nabla f(x^*)^T d \geq 0.$$

5.8 Notes and References

The conditions given here are classical *local* conditions in nonlinear programming. Early references and historical notes can be found, for instance, in the books [92, 179] and [12]. These books are also useful references for the study of constraint qualifications.

The penalty function approach of McShane followed here for deriving the FJ conditions, makes use only of elementary calculus and hence it is employed in various more recent text books for simplifying the study of the optimality conditions. See, in particular, [16] and [109], on which this chapter is based.

Chapter 6

Duality Theory



In this chapter we give a short introduction to *duality theory*, which plays an important role both in the theoretical analysis of optimization problems and in the development of computational algorithms. In particular, after a short discussion on the general motivation of duality, we give some basic result on *Lagrangian duality* and, under appropriate assumptions, we obtain new interpretations of the KKT multipliers. We also present the Wolfe's duality theory, which is at the basis of important classes of machine learning models and algorithms.

6.1 The Dual Problem

Consider the minimization problem (which will be called *primal problem*):

$$\begin{array}{ll} \min & f(x) \\ x \in S & \end{array} \quad (\text{P})$$

where $S \subseteq R^n$. If $S = \emptyset$ we set $\inf_{x \in S} f(x) = +\infty$.

The basic idea of duality theory is that of constructing, in correspondence to problem (P), a new problem, known as the *dual problem*, such that this problem, under appropriate conditions, can give useful information on problem (P). In particular, we define a dual problem of the form

$$\begin{array}{ll} \max & \psi(u) \\ u \in U & \end{array} \quad (\text{D})$$

in a way that (at least) the following property, known as *weak duality*, is satisfied

$$\inf_{x \in S} f(x) \geq \sup_{u \in U} \psi(u). \quad (6.1)$$

Here again we admit the possibility that U is empty and hence, in this case, we set $\sup_{u \in U} \psi(u) = -\infty$.

If both feasible sets are non empty, we obviously have, in case of weak duality, that

$$f(x) \geq \psi(u), \quad \text{for all } x \in S, u \in U.$$

When weak duality holds we can obtain some useful characterizations of the solutions of problem (P). In particular, we can deduce:

- (a) bounds on the optimal value;
- (b) sufficient optimality conditions.

As regards the first point, suppose there exists an optimal solution $x^* \in S$ of Problem (P), so that the optimal value f^* satisfies

$$f^* = f(x^*) = \min_{x \in S} f(x) = \inf_{x \in S} f(x).$$

By definition, an upper bound on the optimal value f^* can be obtained if we know a feasible point $\bar{x} \in S$, as we obviously have $f^* \leq f(\bar{x})$. If weak duality holds and a dual feasible point \bar{u} exists, we must have

$$f(\bar{x}) \geq f^* \geq \sup_{u \in U} \psi(u) \geq \psi(\bar{u}),$$

which yields also a lower bound on the optimal value. Thus, if $f(\bar{x}) - \psi(\bar{u})$ is sufficiently small (in comparison with $|f(\bar{x})|$) we can consider \bar{x} as a good approximation of the (unknown) optimal solution x^* .

If weak duality holds, it easily seen that the condition

$$f(\bar{x}) = \psi(\bar{u}), \quad \bar{x} \in S, \quad \bar{u} \in U,$$

is a sufficient condition for having that \bar{x} and \bar{u} are optimal solutions, respectively, of (P) and (D). In fact, recalling weak duality, we have

$$f(\bar{x}) = \psi(\bar{u}) \leq f(x) \quad \text{for all } x \in S,$$

which implies that \bar{x} is an optimal solution of (P). Similarly,

$$\psi(\bar{u}) = f(\bar{x}) \geq \psi(u) \quad \text{for all } u \in U,$$

which implies that \bar{u} is an optimal solution of (D).

If $\inf_{x \in S} f(x) = -\infty$, we have necessarily that the dual feasible set is empty. Similarly, if $\sup_{u \in U} \psi(u) = +\infty$, the primal feasible set must be empty.

For some important classes of problems, such as, for instance, linear programming and many convex problems, we can construct a dual problem such that the

following stronger condition, called *strong duality*, is satisfied.

$$\inf_{x \in S} f(x) = \sup_{u \in U} \psi(u). \quad (6.2)$$

In this case the condition

$$f(\bar{x}) = \psi(\bar{u}), \quad \bar{x} \in S, \quad \bar{u} \in U, \quad (6.3)$$

is a *necessary and sufficient* optimality condition for \bar{x} and \bar{u} .

The problems for which it is possible to establish strong duality are problems, typically convex, that admit a good characterization of optimality. In some cases the dual problem is simpler than the primal and this can be used in computational techniques (see, e.g., the training problem of linear Support Vector Machine later analyzed). In other cases algorithms for constrained problems make use of both primal and dual information (*primal-dual algorithms*).

When condition (6.2) cannot be satisfied, so that

$$\inf_{x \in S} f(x) > \sup_{u \in U} \psi(u). \quad (6.4)$$

we say that there exists a “*duality gap*”.

The construction of a (useful) dual problem can be performed by employing different criteria. In the next subsection we will confine ourselves to introduce a dual problem based on the Lagrangian function.

6.2 Lagrangian Duality

6.2.1 Basic Concepts, Weak and Strong Lagrangian Duality

Let us consider, without loss of generality, a primal problem of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \quad x \in X \end{aligned} \quad (\text{P})$$

where $x \in R^n$, $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$ and $X \subseteq R^n$ is some unspecified non empty set. Then the feasible set is given by: $S = \{x \in X : g(x) \leq 0\}$. In correspondence to Problem (P) we can construct the Lagrangian function

$$L(x, u) = f(x) + u^T g(x), \quad u \in R^m$$

and we can define on the extended real space $R \cup \{\pm\infty\}$ the function ψ by letting

$$\psi(u) = \inf_{x \in X} L(x, u).$$

Now we define the set $U = \{u \in R^m : u \geq 0\}$ and we consider the problem

$$\begin{aligned} & \max \psi(u) \\ & u \in U. \end{aligned} \quad (\text{D})$$

In the next proposition we show that Problem (D) can be taken as the dual of (P).

Proposition 6.1 (Weak Duality) *Consider the problems (P) and (D) defined above. Then weak duality holds, that is*

$$\inf_{x \in S} f(x) \geq \sup_{u \in U} \psi(u). \quad (6.5)$$

Proof Suppose first that S is non empty and that $\inf_{x \in S} f(x) > -\infty$. Then, for every $x \in S$ and every $u \geq 0$, as $g(x) \leq 0$, we have $u^T g(x) \leq 0$ so that

$$f(x) \geq L(x, u) = f(x) + u^T g(x) \geq \inf_{x \in X} L(x, u) = \psi(u)$$

and the assertion holds. If $\inf_{x \in S} f(x) = -\infty$ then we have also $\inf_{x \in X} L(x, u) = -\infty$.

Finally if S is empty, we have, by convention, $\inf_{x \in S} f(x) = +\infty$ and we can write again (6.5). \square

We can prove that the dual objective $\psi(u)$ is a *concave* function.

Proposition 6.2 *The dual objective $\psi(u) = \inf_{x \in X} L(x, u)$ is a concave function.*

Proof Let $u_1, u_2 \in U$ and $\tau \in [0, 1]$. We can write

$$\begin{aligned} \psi((1-\tau)u_1 + \tau u_2) &= \inf_{x \in X} L(x, (1-\tau)u_1 + \tau u_2) \\ &= \inf_{x \in X} [f(x) + ((1-\tau)u_1 + \tau u_2)^T g(x)] \\ &= \inf_{x \in X} [\tau f(x) + (1-\tau)f(x) + (1-\tau)u_1^T g(x) \\ &\quad + \tau u_2^T g(x)] \\ &= \inf_{x \in X} [(1-\tau)L(x, u_1) + \tau L(x, u_2)] \\ &\geq \inf_{x \in X} [(1-\tau)L(x, u_1)] + \inf_{x \in X} \tau L(x, u_2) \\ &= (1-\tau)\psi(u_1) + \tau\psi(u_2), \end{aligned}$$

and this concludes the proof. \square

Then the dual problem (D) is a convex programming problem, where we *maximize* a *concave function* over the convex set U .

The preceding concepts and results can be extended easily to primal problems that include also equality constraints, of the form

$$\begin{aligned} \min f(x) \\ h(x) = 0, \quad (\text{P}) \\ g(x) \leq 0, x \in X. \end{aligned}$$

where now $h : R^n \rightarrow R^p$ represents the vector of equality constraints. In correspondence to Problem (P) above, we can construct the Lagrangian function

$$L(x, u, v) = f(x) + u^T g(x) + v^T h(x), \quad u \in R^m$$

and we can define the function $\phi : R^m \rightarrow R \cup \{-\infty\}$ by letting

$$\phi(u, v) = \inf_{x \in X} L(x, u, v).$$

Then we can construct the dual problem in the variables $u \in R^m, v \in R^p$, with v unrestricted.

$$\begin{aligned} \max \phi(u, v) \\ u \geq 0. \quad (\text{D}) \end{aligned}$$

It is easily verified that weak duality holds.

Consider now an important example of construction of the dual problem. We consider a LP problem of the form

$$\begin{aligned} \min c^T x, \quad (\text{P}) \\ Ax \geq b, \end{aligned}$$

which we take as primal problem. The Lagrangian function is given by

$$L(x, u) = c^T x + u^T (b - Ax),$$

and hence, taking $X = R^n$, the dual function is given by

$$\psi(u) = \inf_{x \in R^n} L(x, u) = \inf_{x \in R^n} b^T u + (A^T u - c)^T x = \begin{cases} -\infty, & \text{if } A^T u - c \neq 0 \\ b^T u, & \text{if } A^T u - c = 0 \end{cases}.$$

Then the dual problem can be defined as

$$\begin{aligned} \max b^T u \\ A^T u = c, u \geq 0. \quad (\text{D}) \end{aligned}$$

Recalling Proposition 5.14, we know, from the KKT conditions, that the necessary and sufficient optimality condition for x^* to be a global minimum point of Problem (P) is that there exists a KKT multiplier $\lambda^* \in R^m$ such that:

- (a) $\nabla_x L(x^*, \lambda^*) \equiv c - A^T \lambda^* = 0, \quad \lambda^* \geq 0;$
- (b) $Ax^* \geq b;$
- (c) $b^T \lambda^* = c^T x^*.$

This implies that $u^* = \lambda^*$ can be interpreted as a solution of the dual problem and hence that strong duality holds.

In the general case, strong duality cannot be established unless suitable assumptions are introduced. Here we will confine ourselves to show that, under suitable convexity and differentiability assumptions, a pair (x^*, λ^*) satisfying the KKT conditions, is a primal-dual optimal solution that guarantees strong duality.

More specifically, consider the primal problem

$$\begin{aligned} \min & f(x) \\ & g(x) \leq 0, \end{aligned} \quad (\text{P})$$

with associated dual problem

$$\begin{aligned} \max & \psi(u) \\ & u \geq 0, \end{aligned} \quad (\text{D})$$

with $\psi(u) = \inf_{x \in R^n} L(x, u)$. Then we can establish the following result.

Proposition 6.3 (Strong Duality) Suppose that the functions f and $g_i, i = 1, \dots, m$ are continuously differentiable and convex. Suppose that (x^*, λ^*) is a KKT pair for problem (P). Then x^* and λ^* are optimal solutions, respectively, for the primal and the dual problem, with no duality gap.

Proof Suppose that (x^*, λ^*) is a KKT pair for problem (P) above, so that we have

- (a) $\nabla_x L(x^*, \lambda^*) \equiv \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) = 0, \quad \lambda^* \geq 0;$
- (b) $g(x^*) \leq 0;$
- (c) $\lambda_i^* g_i(x^*) = 0, \quad i = 1, \dots, m.$

As $L(x, \lambda^*)$, with $\lambda^* \geq 0$, is a convex function of x , condition (a) implies that x^* (which is, by assumption, a stationary point of $L(x, \lambda^*)$) must be a solution of the problem $\inf_{x \in R^n} L(x, \lambda^*)$, so that $L(x^*, \lambda^*) = \inf_{x \in R^n} L(x, \lambda^*) = \psi(\lambda^*)$ with $\lambda^* \geq 0$. Moreover, by assumptions (b) and (c), we have $\lambda^* g(x^*) = 0$ and hence we obtain $L(x^*, \lambda^*) = f(x^*)$, so that $f(x^*) = \psi(\lambda^*)$. As weak duality holds by construction, this implies that also strong duality holds and that x^* and λ^* are solutions of (P) and (D). \square

6.2.2 Saddle Point Optimality Conditions

Consider the problem

$$\begin{aligned} \min f(x) \\ g(x) \leq 0, x \in X \subseteq \mathbb{R}^n. \end{aligned} \quad (\text{P})$$

and the associated Lagrangian function

$$L(x, u) = f(x) + u^T g(x),$$

where $x \in \mathbb{R}^n$, $u \in U = \{u \in \mathbb{R}^m : u \geq 0\}$. We introduce the following definition.

Definition 6.1 A point $(\hat{x}, \hat{u}) \in X \times U$ is said to be a *saddle point* of L if, for all points $(x, u) \in X \times U$, we have

$$L(\hat{x}, u) \leq L(\hat{x}, \hat{u}) \leq L(x, \hat{u}).$$

□

It can be shown that the existence of a saddle point is a necessary and sufficient condition for the existence of a primal-dual optimal pair with no duality gap. The following well-known results are reported in the special case of inequality constrained problem (P) and the proofs are given in the appendix to this chapter, for convenience of the reader.

The extension to problems including equality constraints can be found in the literature (see, in particular, Theorems 6.2.5 and 6.2.6 in [12], which we have followed here).

Proposition 6.4 (Saddle Point Condition) A point $(\hat{x}, \hat{u}) \in X \times U$ is a saddle point for the Lagrangian $L(x, u)$ if and only if

- (a) $L(\hat{x}, \hat{u}) = \min_{x \in X} L(x, \hat{u})$,
- (b) $g(\hat{x}) \leq 0$,
- (c) $\hat{u}^T g(\hat{x}) = 0$.

Moreover, (\hat{x}, \hat{u}) is a saddle point if and only if \hat{x} and \hat{u} are the optimal solutions, respectively, of the primal and the dual problem with no duality gap. □

The condition given above do not impose neither convexity assumptions nor differentiability requirements. The relationships between saddle point conditions and KKT conditions are given in the next proposition.

Proposition 6.5 (Saddle Point Conditions and KKT Conditions) *Let $x^* \in X \subseteq \mathbb{R}^n$ and suppose that f and g_i for $i \in I_0(x^*)$ are convex and continuously differentiable in a neighborhood of x^* , where $I_0(x^*) = \{i : g_i(x^*) = 0\}$. Suppose that (x^*, λ^*) is a KKT pair; i.e., we have*

- (a) $\nabla_x L(x^*, \lambda^*) \equiv \nabla f(x^*) + \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*) = 0, \quad \lambda^* \geq 0$;
- (b) $g(x^*) \leq 0$;
- (c) $\lambda_i^* g_i(x^*) = 0, \quad i = 1, \dots, m$.

Then (x^, λ^*) is a saddle point for L .*

Conversely, if (\hat{x}, \hat{u}) is a saddle point, with $\hat{x} \in \text{int}(X)$ and $\hat{u} \geq 0$, then \hat{x} is a feasible point for Problem (P), and (\hat{x}, \hat{u}) satisfies the KKT conditions specified by (a) (b) (c). \square

6.3 Wolfe's Dual

As seen, Lagrangian duality has been introduced in a general setting without assumptions on the objective function and on the functions defining the feasible set. This leads to a dual function, to be maximized, defined as the infimum with respect to the primal variables of the Lagrangian function. Under convexity and differentiability assumptions on the involved functions it is possible to characterize the dual problem taking into account that the infimum is obtained by imposing that the gradient of the Lagrangian function, with respect to the primal variables, is equal to zero.

Consider the primal problem

$$\begin{aligned} & \min f(x) \\ & g(x) \leq 0 \end{aligned} \tag{6.6}$$

assume that the functions f and $g_i, i = 1, \dots, m$ are continuously differentiable and convex. The possible presence of further linear equality constraints can be easily managed, but it is not considered here for sake of brevity.

As we know, the Lagrangian function is

$$L(x, u) = f(x) + u^T g(x),$$

with $u \geq 0$. By the convexity assumptions, given $u \geq 0$ and any point $\xi \in R^n$ such that

$$\nabla_x L(\xi, u) = 0,$$

we have $\xi \in \arg \min_x L(x, u)$, which implies that the dual function $\psi(u) = \inf_{x \in R^n} L(x, u)$ takes the value $L(\xi, u)$. Therefore, we can consider the dual problem, called *Wolfe's dual*, defined as follows

$$\begin{aligned} & \max_{x, u} L(x, u) \\ & \nabla_x L(x, u) = 0 \\ & u \geq 0. \end{aligned} \tag{6.7}$$

Then, the Wolfe's dual has the pair (x, u) as dual variables. We show that any KKT pair (x^*, λ^*) for problem (6.6) is a solution of Wolfe's dual (6.7) (in the general case, the viceversa is not true).

Proposition 6.6 Assume that problem (6.6) admits an optimal solution x^* and that there exists a vector $\lambda^* \in R^m$ such that (x^*, λ^*) is a KKT pair for problem (6.6). Then (x^*, λ^*) is a solution of the Wolfe's dual (6.7) and we have $f(x^*) = L(x^*, \lambda^*)$.

Proof The KKT conditions imply that $g(x^*) \leq 0$ and that

$$\begin{aligned} & \nabla_x L(x^*, \lambda^*) = 0, \\ & (\lambda^*)^T g(x^*) = 0, \\ & \lambda^* \geq 0. \end{aligned} \tag{6.8}$$

Therefore, the point (x^*, λ^*) is a feasible point for problem (6.7), and we have $f(x^*) = L(x^*, \lambda^*)$. We show that (x^*, λ^*) is a solution of (6.7).

Let (x, λ) be a feasible point of problem (6.7), i.e., we have $\nabla_x L(x, \lambda) = 0$ and $\lambda \geq 0$. Function $L(x, \lambda)$ is a convex function of x . Indeed, $f(x)$ is convex, the second term is a linear combination of convex functions with nonnegative coefficients λ_i and hence is convex, so that $L(x, \lambda)$ is a convex function of x . Then, for any $x, y \in R^n$, we can write

$$L(y, \lambda) \geq L(x, \lambda) + \nabla_x L(x, \lambda)^T (y - x). \tag{6.9}$$

From (6.8), taking into account that $g(x^*) \leq 0$ and $\lambda \geq 0$, using (6.9) and the condition $\nabla_x L(x, \lambda) = 0$, for any $\lambda \geq 0$ we can write

$$\begin{aligned} L(x^*, \lambda^*) &= f(x^*) \geq f(x^*) + \sum_{i=1}^m \lambda_i g_i(x^*) = L(x^*, \lambda) \\ &\geq L(x, \lambda) + \nabla_x L(x, \lambda)^T (x^* - x) = L(x, \lambda) \end{aligned}$$

and hence the thesis is proved. \square

As already mentioned, in the general case, given a solution (\bar{x}, \bar{u}) of Wolfe's dual, we can not draw conclusions with respect to the primal problem (6.6). In the particular case of convex quadratic programming problems with linear constraints we can state the following result.

Proposition 6.7 Consider the problem

$$\begin{aligned} \min_x f(x) &= \frac{1}{2} x^T Q x + c^T x \\ Ax - b &\leq 0, \end{aligned} \tag{6.10}$$

where $Q \in R^{n \times n}$, $c \in R^n$, $A \in R^{m \times n}$, $b \in R^m$. Suppose that the matrix Q is symmetric and positive semidefinite. Let (\bar{x}, \bar{u}) be a solution of Wolfe's dual (6.7). Then, there exists a vector x^* (not necessarily equal to \bar{x}) such that

- (i) $Q(x^* - \bar{x}) = 0$;
- (ii) x^* is a solution of problem (6.10); and
- (iii) x^* is a global minimum point of (6.10) with associated multiplier \bar{u} .

Proof Consider the Wolfe's dual

$$\max_{x,u} \frac{1}{2} x^T Q x + c^T x + u^T (Ax - b)$$

$$Qx + A^T u + c = 0$$

$$u \geq 0$$

The constraint $Qx + A^T u + c = 0$ implies

$$x^T Qx + c^T x + u^T Ax = 0. \tag{6.11}$$

Using (6.11), the dual problem can be written as follows

$$\begin{aligned} \min_{x,u} & \frac{1}{2}x^T Qx + u^T b \\ Qx + A^T u + c &= 0 \\ u &\geq 0 \end{aligned} \tag{6.12}$$

Let (\bar{x}, \bar{u}) be a solution of (6.12). Consider the Lagrangian function associated to problem (6.12)

$$W(x, u, v, z) = \frac{1}{2}x^T Qx + u^T b - v^T(Qx + A^T u + c) - z^T u.$$

Since (\bar{x}, \bar{u}) is solution of Wolfe's dual (6.7), from the KKT conditions we have that there exist a vector $\bar{v} \in R^n$ and a vector $\bar{z} \in R^m$ such that

$$\begin{aligned} \nabla_x W &= Q\bar{x} - Q\bar{v} = 0 \\ \nabla_u W &= b - A\bar{v} - \bar{z} = 0 \\ Q\bar{x} + A^T \bar{u} + c &= 0 \\ \bar{z}^T \bar{u} &= 0 \\ \bar{z} &\geq 0 \\ \bar{u} &\geq 0 \end{aligned} \tag{6.13}$$

From the second and fifth relations we get $\bar{z} = b - A\bar{v} \geq 0$, and hence the preceding conditions can be rewritten as follows

$$\begin{aligned} Q\bar{x} - Q\bar{v} &= 0 \\ -b + A\bar{v} &\leq 0 \\ Q\bar{x} + A^T \bar{u} + c &= 0 \\ -\bar{u}^T b + \bar{u}^T A\bar{v} &= 0 \\ \bar{u} &\geq 0 \end{aligned} \tag{6.14}$$

By subtracting the first condition from the third condition we have

$$Q\bar{v} + A^T\bar{u} + c = 0, \quad (6.15)$$

thus obtaining that the pair (\bar{v}, \bar{u}) satisfies the following conditions

$$A\bar{v} - b \leq 0$$

$$Q\bar{v} + A^T\bar{u} + c = 0$$

$$\bar{u}^T(A\bar{v} - b) = 0$$

$$\bar{u} \geq 0$$

Now consider problem (6.10). Conditions KKT hold since the feasible set is defined by linear inequalities. The above conditions, satisfied by the pair (\bar{v}, \bar{u}) , are the KKT conditions for problem (6.10). Taking into account the convexity of the objective function f , being the Hessian matrix Q positive semidefinite, the KKT conditions are sufficient, in this case, to ensure that \bar{v} is a solution of problem (6.10).

Therefore, \bar{v} is a global minimum point of (6.10) with associated multiplier \bar{u} . Setting $x^* = \bar{v}$ we have that x^* is solution of problem (6.10); furthermore, taking into account the first relation of (6.14), we have

$$Qx^* = Q\bar{v} = Q\bar{x}.$$

Then assertions (i)–(iii) are proved. \square

6.3.1 Wolfe's Dual in Quadratic Convex Programming Problems with Linear Inequality Constraints

An important application of the Wolfe's dual concerns the training problem of Support Vector Machine (SVM), which is widely used as a simple and efficient machine learning model for linear and nonlinear classification as well as for regression problems [254]. The primal problem in SVM training is given by a convex quadratic problem with a number of linear inequalities equal to the number of training observations. It can be shown that its Wolfe's dual is defined by a convex quadratic minimization problem (only in the dual variables) with a single linear equality constraint and “simple” box constraints.

Without going into the details of SVM theory, we consider here a simplified primal problem of the following form

$$\begin{aligned} \min f(x) &= \frac{1}{2} \|x\|^2 \\ a_i^T x - 1 &\geq 0 \quad i = 1, \dots, m \end{aligned} \tag{6.16}$$

where $a_i \in R^n$ for $i = 1, \dots, m$. We assume that the feasible set of (6.16) is not empty. Then, as the objective function is coercive and strictly convex, the solution of (6.16) exists and is unique.

The Lagrangian function associated to (6.16) is the following function

$$L(x, \lambda) = \frac{1}{2} \|x\|^2 - \sum_{i=1}^m \lambda_i (a_i^T x - 1) \tag{6.17}$$

The Wolfe's dual of (6.16) is

$$\max L(x, \lambda) = \frac{1}{2} \|x\|^2 - \sum_{i=1}^m \lambda_i (a_i^T x - 1)$$

$$\nabla_x L(x, \lambda) = 0$$

$$\lambda \geq 0,$$

that is,

$$\begin{aligned} \max L(x, \lambda) &= \frac{1}{2} \|x\|^2 - \sum_{i=1}^m \lambda_i (a_i^T x - 1) \\ x &= \sum_{i=1}^m \lambda_i a_i \end{aligned} \tag{6.18}$$

$$\lambda \geq 0,$$

which can be equivalently rewritten as follows

$$\min \Gamma(\lambda) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m a_i^T a_j \lambda_i \lambda_j - \sum_{i=1}^m \lambda_i \tag{6.19}$$

$$\lambda_i \geq 0 \quad i = 1, \dots, m$$

Then, setting $A = [a_1, \dots, a_m]$, $\lambda^T = [\lambda_1, \dots, \lambda_m]$, Wolfe's dual of (6.16) is a *convex quadratic programming problem* (only in the dual variables) of the form

$$\begin{aligned} \min \Gamma(\lambda) &= \frac{1}{2} \lambda^T A^T A \lambda - \mathbf{1}^T \lambda \\ &\quad \lambda \geq 0 \end{aligned} \tag{6.20}$$

where $\mathbf{1}^T = [1, \dots, 1]$. Propositions 6.6 implies that problem (6.20) admits solution. Given any solution λ^* of (6.20), using Proposition 6.7 and recalling the equivalence between problems (6.20) and (6.18) it follows that the unique solution of (6.16) is

$$x^* = \sum_{i=1}^m \lambda_i^* a_i.$$

By comparing the primal problem (6.16) and its Wolfe's dual (6.20), whose feasible set is defined by “simple” nonnegative constraints, we observe that it may be convenient to solve Wolfe's dual, instead of the primal problem, when the number m of inequalities is large, and this happens, for instance, in many real applications of SVM.

6.4 Appendix

Proof of Proposition 6.4 A point $(\hat{x}, \hat{u}) \in X \times U$ is a saddle point for the Lagrangian $L(x, u)$ if and only if

- (a) $L(\hat{x}, \hat{u}) = \min_{x \in X} L(x, \hat{u})$,
- (b) $g(\hat{x}) \leq 0$,
- (c) $\hat{u}^T g(\hat{x}) = 0$.

Moreover, (\hat{x}, \hat{u}) is a saddle point if and only if \hat{x} and \hat{u} are the optimal solutions, respectively, of the primal and the dual problem with no duality gap. \square

Proof Let $(\hat{x}, \hat{u}) \in X \times U$ be a saddle point. Then (a) holds by definition. The definition of saddle point implies also that

$$L(\hat{x}, \hat{u}) = f(\hat{x}) + \hat{u}^T g(\hat{x}) \geq L(\hat{x}, u) = f(\hat{x}) + u^T g(\hat{x}), \quad (6.21)$$

which implies, in turn, that $\hat{u}^T g(\hat{x}) \geq u^T g(\hat{x})$. Hence, as $\hat{u}^T g(\hat{x}) \leq 0$, we have also that $u^T g(\hat{x}) \leq 0$, for every vector $u \geq 0$. Then, we must have $g(\hat{x}) \leq 0$ for, otherwise, if $g_j(\hat{x}) > 0$ for some j , we could choose $u \in U$ such that $u = \alpha e_j$, where e_j is the j_{th} column of the identity matrix I_m and this would contradict the inequality $u^T g(\hat{x}) \leq 0$, for sufficiently large $\alpha > 0$. This establishes (b). Now, if we take $u = 0$ in (6.21), we have $\hat{u}^T g(\hat{x}) \geq 0$ and hence, as we have also $\hat{u}^T g(\hat{x}) \leq 0$, we must have $\hat{u}^T g(\hat{x}) = 0$ and this proves (c).

Now suppose that $(\hat{x}, \hat{u}) \in X \times U$ is a point satisfying properties (a)(b)(c). It follows directly from (a) that

$$L(\hat{x}, \hat{u}) \leq L(x, \hat{u}),$$

for all $x \in X$. Moreover, by (b) we have that

$$L(\hat{x}, \hat{u}) = f(\hat{x}) + \hat{u}^T g(\hat{x}) = f(\hat{x}) \geq f(\hat{x}) + u^T g(\hat{x}),$$

for all $u \geq 0$. Then we can assert that (\hat{x}, \hat{u}) is a saddle point with.

Finally, suppose again that $(\hat{x}, \hat{u}) \in X \times U$ is a saddle point. Then (a)(b)(c) must be true, as we have shown, and hence, by (b) the point \hat{x} is feasible point and we have also that the dual objective function satisfies $\psi(\hat{u}) = \inf_{x \in X} L(x, \hat{u}) = f(\hat{x})$ and hence, as weak duality holds we have that \hat{x} and \hat{u} are optimal primal-dual solutions with zero duality gap.

Now suppose that \hat{x} and \hat{u} are optimal primal-dual solutions with zero duality gap. Then

$$\psi(\hat{u}) = \min\{f(x) + \hat{u}^T g(x), x \in X\} \leq f(\hat{x}) + \hat{u}^T g(\hat{x}) \leq f(\hat{x}).$$

By assumption we have $\psi(\hat{u}) = f(\hat{x})$ This implies that (a)–(c) are satisfied and hence that $(\hat{x}, \hat{u}) \in X \times U$ is a saddle point. \square

Proof Let $x^* \in X \subseteq R^n$ and suppose that f and g_i for $i \in I_0(x^*)$ are convex and continuously differentiable in a neighborhood of x^* , where $I_0(x^*) = \{i : g_i(x^*) = 0\}$. Suppose that (x^*, λ^*) is a KKT pair, i.e., we have

- (a) $\nabla_x L(x^*, \lambda^*) \equiv \nabla f(x^*) + \sum_{i \in I_0(x^*)} \lambda_i^* \nabla g_i(x^*) = 0, \quad \lambda^* \geq 0;$
- (b) $g(x^*) \leq 0;$
- (c) $\lambda_i^* g_i(x^*) = 0, \quad i = 1, \dots, m.$

(continued)

Then (x^*, λ^*) is a saddle point for L .

Conversely, if (\hat{x}, \hat{u}) is a saddle point, with $\hat{x} \in \text{int}(X)$ and $\hat{u} \geq 0$, then \hat{x} is a feasible point for Problem (P), and (\hat{x}, \hat{u}) satisfies the KKT conditions specified by (a)–(c). \square

Proof Suppose that $(x^*, \lambda^*) \in X \times U$ satisfies KKT conditions specified by (a) (b) (c). By the convexity assumptions, we have, for all $x \in X$:

$$f(x) \geq f(x^*) + \nabla f(x^*)^T(x - x^*), \quad (6.22)$$

$$g_i(x) \geq g_i(x^*) + \nabla g_i(x^*)^T(x - x^*), \quad \text{for all } i \in I_0(x^*). \quad (6.23)$$

Multiplying each of the inequalities (6.23) by λ_i^* , for all $i \in I_0(x^*)$ and summing to (6.22), taking into account the KKT conditions, we obtain

$$L(x, \lambda^*) \geq L(x^*, \lambda^*). \quad (6.24)$$

Moreover, as $\lambda^* \geq 0$, $(\lambda^*)^T g(x^*) = 0$ and $g(x^*) \leq 0$, we have

$$L(x^*, \lambda^*) = f(x^*) \geq f(x^*) + \lambda^T g(x^*) = L(x^*, \lambda), \quad (6.25)$$

for all $\lambda \geq 0$. Thus, by (6.24) and (6.25), we have that (x^*, λ^*) is a saddle point. Conversely, let $(\hat{x}, \hat{u}) \in \text{int}(X) \times U$ be a saddle point for L . By Proposition 6.5, we have that conditions (a)(b)(c) stated there must hold. Moreover, as, by assumption, $x \in \text{int}(X)$ and L is a continuously differentiable and convex function of x on a neighborhood of $\hat{x} \in \text{int}(X)$, condition (a) implies that $\nabla_x L(\hat{x}, \hat{u}) = 0$ so that (\hat{x}, \hat{u}) satisfies KKT conditions specified by (a)–(c). \square

6.5 Exercises

6.1 Using the Lagrangian duality show that the dual of the problem

$$\begin{aligned} & \min c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

is the following problem

$$\begin{aligned} & \max b^T u \\ & A^T u \leq c. \end{aligned}$$

6.2 Using the duality theory prove the Farkas' Lemma.

Let $A \in R^{m \times n}$, $b \in R^m$. Then exactly one of the following conditions hold:

(I) *there exists $x \in R^n$ such that*

$$Ax = b \quad x \geq 0;$$

(II) *there exists $y \in R^m$ such that*

$$b^T y > 0 \quad A^T y \leq 0.$$

6.3 Consider the problem

$$\begin{aligned} \min f(x) &= \frac{1}{2} \|x\|^2 + C \sum_{i=1}^m \xi_i \\ a_i^T x - 1 + \xi_i &\geq 0 \quad i = 1, \dots, m \\ \xi &\geq 0. \end{aligned}$$

Show that its Wolfe's dual is the following problem

$$\begin{aligned} \min \Gamma(\lambda) &= \frac{1}{2} \lambda^T A^T A \lambda - 1^T \lambda \\ 0 \leq \lambda &\leq C. \end{aligned}$$

6.6 Notes and References

Duality is a powerful tool in optimization both from a theoretical and an algorithmic point of view. As reported in [10], the duality for linear programming was foreseen by J. von Neumann and G.B. Dantzig in 1947. The book [19] is a classical reference for duality in linear programming, as well as duality for nonlinear programming is deeply investigated in the classical books [12, 16, 229]. Finally, we remark that Support Vector Machine (SVM), a popular and widely used machine learning model presented in the mid 1990s, and the related kernel-based learning strongly exploits the Wolfe's dual theory. The theory of SVM, of kernel learning and their relationship with Wolfe's dual are analyzed in [254]. In this chapter, the matter of Wolfe's dual has been inspired by the books [93] and [179].

Chapter 7

Optimality Conditions Based on Theorems of the Alternative



In this chapter we prove again Fritz John (FJ) conditions (and then KKT conditions, under constraint qualifications) following the geometric approach based on *theorems of the alternative*, i.e., theorems stating that exactly one of two systems has solution. These latter are reported in the appendix where, in particular, optimality conditions for linear programming are proved using only an algebraic approach. We first consider problems with only inequality constraints and we give a characterization of the feasible directions in the general, non convex, case. Then, for problems with nonlinear equality and inequality constraints, we introduce the notion of *tangent directions*, which can define a feasible path even when feasible directions do not exist at the minimizer. By characterizing feasible directions, tangent directions and descent directions using linear approximations of the constraints, we can reobtain the Fritz John conditions in the general case, by employing the theorems of the alternative.

7.1 Introduction

In Chap. 5 we established Fritz John (FJ) conditions (and then KKT conditions, under constraint qualifications) by employing a *penalty approach*. This has permitted to obtain FJ conditions through a single proposition, without requiring, in the general case, the use of other basic concepts and results, such as, for instance, the *implicit function theorem* of Real Analysis.

An alternative, more “classical”, approach was followed in Chap. 4 with reference to problems with convex feasible set. Necessary optimality conditions were expressed by requiring that, at a local minimizer *there not exist feasible directions that are descent directions*.

The extension of this result to the general non convex case is not straightforward, since the characterization of feasible directions may be not simple and, moreover,

especially in the presence of nonlinear equality constraints, we may have situations where do not exist feasible directions at non critical points.

As an example, consider the problem

$$\begin{aligned} \min & x_2 \\ & x_2 - x_1^2 = 0, \end{aligned}$$

where the optimal solution is obviously $(0, 0)$. It is easily seen that at any given feasible point (x_1, x_2) we cannot find feasible directions. Therefore, in this case it could be useful to identify feasible curvilinear descent paths at non critical points. This requires the introduction of the concept of *tangent direction*, which is intended as the tangent to a differentiable, possibly curvilinear, feasible path originating from a given feasible point. It will be shown that necessary optimality conditions can be expressed by requiring that at a local minimizer *there not exist tangent directions that are descent directions*.

Using this approach, it is possible to obtain a necessary optimality condition by requiring that a system of linear inequalities and equalities does not admit a solution. As we will see in the sequel, we can construct a linear approximation of the constraints, which yields a system of the form

$$B(x)d < 0, \quad C(x)d \leq 0, \quad D(x)d = 0,$$

where x is feasible point, B, C, D are matrices depending on the problem functions, and we require that does not exist a solution d of this system.

Using the *theorems of the alternative* we can restate these conditions by requiring that an alternative problem has a solution. The solution of the alternative system will be interpreted as the vector of generalized Lagrange multipliers and the solution of the alternative system will yield the FJ conditions.

7.2 Optimality Conditions for Problems with Inequality Constraints

Let us consider problems of the form

$$\begin{aligned} \min & f(x) \\ & g(x) \leq 0, \end{aligned}$$

where $f : R^n \rightarrow R$ and $g : R^n \rightarrow R^m$ are continuously differentiable functions. The feasible set is thus the set $S = \{x \in R^n : g(x) \leq 0\}$.

A characterization of feasible directions at a given points $x \in R^n$ can be given by identifying, whenever possible, active constraints that are *concave* at x , in the sense

of Definition 5.1, that is, by requiring that

$$g_i(y) \leq g_i(x) + \nabla g_i(x)^T (y - x), \quad \text{for all } y \in B(x, \rho),$$

where $B(x, \rho)$, with $\rho > 0$ is a neighborhood of x .

The next proposition is a slightly modified version of the result established in [179].

Proposition 7.1 *Let $x \in S$ and assume that $g : R^n \rightarrow R^m$ is continuously differentiable in a neighborhood of x . Let $I_0(x) = \{i : g_i(x) = 0\}$ and define the index sets*

$$\begin{aligned} V &\subseteq \{i \in I_0(x) : g_i \text{ is concave at } x\} \\ W &= \{i \in I_0(x) : i \notin V\}. \end{aligned}$$

Let $d \in R^n$ a vector such that

$$\begin{aligned} \nabla g_i(x)^T d &< 0 \quad \text{forall } i \in W \\ \nabla g_i(x)^T d &\leq 0 \quad \text{for all } i \in V. \end{aligned}$$

Then d is a feasible direction for S at x .

Proof Let d a vector that satisfies the conditions stated. Preliminarily, we observe that, by continuity, d is feasible for all the constraints g_i such that $i \notin I_0(x)$.

As the number of constraints is finite, there exists $t_1 > 0$ such that

$$g_i(x + td) < 0, \quad \text{for all } t \in (0, t_1] \quad \text{and all } i \notin I_0(x).$$

Consider now the constraints that are active at x and suppose first that $i \in W$. In this case, as $\nabla g_i(x)^T d < 0$, the vector d is a descent direction for g_i at x and hence, as the number of constraints is finite, we can find $t_2 > 0$ such that

$$g_i(x + td) < g_i(x) = 0, \quad \text{for all } t \in (0, t_2] \text{ and all } i \in W.$$

Finally, suppose that $i \in V$, so that g_i is concave at x and $\nabla g_i(x)^T d \leq 0$. Then, for sufficiently small values of t we have

$$g_i(x + td) \leq g_i(x) + t \nabla g_i(x)^T d \leq g_i(x) = 0,$$

and hence we can find $t_3 > 0$ such that

$$g_i(x + td) \leq 0, \quad \text{for all } t \in [0, t_3] \text{ and all } i \in V.$$

It can be concluded that, taking $t^* = \min\{t_1, t_2, t_3\}$, we have

$$g(x + td) \leq 0, \text{ for all } t \in [0, t^*],$$

so that d is a feasible direction. \square

On the basis of the preceding proposition we can establish the following necessary optimality condition.

Proposition 7.2 *Let $x^* \in S$ be a local minimizer of f on S and assume that $f : R^n \rightarrow R$ and $g : R^n \rightarrow R^m$ are continuously differentiable on a neighborhood of x^* . Let $I_0(x^*) = \{i : g_i(x^*) = 0\}$ and define the index sets*

$$\begin{aligned} V &\subseteq \{i \in I_0(x^*) : g_i \text{ is concave at } x^*\} \\ W &= \{i \in I_0(x^*) : i \notin V\}. \end{aligned}$$

Then, there not exist $d \in R^n$ such that

$$\begin{aligned} \nabla f(x^*)^T d &< 0 \\ \nabla g_i(x^*)^T d &< 0 \quad \text{for all } i \in W. \\ \nabla g_i(x^*)^T d &\leq 0 \quad \text{for all } i \in V. \end{aligned}$$

Proof The proof follows immediately from the preceding proposition. In fact, if the assertion were false, the direction d would be a feasible descent direction at x^* , and this would contradict the assumption that x^* is a local minimizer. \square

Remark 7.1 Note that we can assume, in particular, $V = \emptyset$ and hence the optimality conditions reduces to imposing that the system

$$\nabla f(x^*)^T d < 0 \quad \nabla g_i(x^*)^T d < 0 \quad \text{for all } i \in I_0(x^*)$$

has no solution. \square

Using the theorems of the alternative, from the preceding results we can easily obtain the Fritz John conditions for the case considered above. This will be shown in the more general case when we have also nonlinear equality constraints and feasible directions may not exist at x^* . We need the introduction of the concept of *tangent direction*.

7.3 Optimality Conditions Based on Tangent Directions

We introduce the following definition [227].

Definition 7.1 (Tangent Direction) A non zero vector $d \in R^n$ is said to be a *tangent direction* for S at $x \in S$ if there exist a sequence of positive numbers $\{t_k\}$, with $t_k \rightarrow 0$, and a sequence of points $x_k \in S$ (converging to x) such that:

$$d = \lim_{k \rightarrow \infty} \frac{x_k - x}{t_k}.$$

□

Equivalently, we can say:

A non zero vector $d \in R^n$ is said to be a tangent direction for S at $x \in S$ if there exist a sequence of positive numbers $\{t_k\}$, with $t_k \rightarrow 0$, and a sequence of points $x_k \in S$ such that:

$$x_k = x + t_k d + r(x, t_k) \quad \text{with} \quad \lim_{k \rightarrow \infty} r(x, t_k)/t_k = 0.$$

□

It is easily seen that a feasible direction is also a tangent direction such that $r(x, t) \equiv 0$ for $t \in [0, \bar{t}]$ for some $\bar{t} > 0$.

Now we can establish a necessary optimality condition in terms of tangent directions, which can be viewed as an extension of Proposition 4.1.

Proposition 7.3 Suppose that $f : R^n \rightarrow R$ is continuously differentiable on a neighborhood of a local minimizer $x^* \in S$ for the problem

$$\min f(x), \quad x \in S.$$

Then there not exist a tangent direction d at x^* such that $\nabla f(x^*)^T d < 0$.

Proof Let $x^* \in S$ be a local minimizer and let $d \in R^n$ be a tangent direction for S at x^* . Reasoning by contradiction, suppose that $\nabla f(x^*)^T d < 0$.

By definition of tangent direction we can find a sequence of positive numbers $\{t_k\}$, with $t_k \rightarrow 0$, and a sequence of points $x_k \in S$ such that:

$$x_k = x^* + t_k \left(d + \frac{r(x^*, t_k)}{t_k} \right).$$

Then, as $r(x^*, t_k)/t_k \rightarrow 0$, taking limits we have that $x_k \rightarrow x^*$. Using the Theorem of the Mean, we can write

$$f(x_k) = f(x^*) + t_k \nabla f(z_k)^T \left(d + \frac{r(x^*, t_k)}{t_k} \right), \quad (7.1)$$

where

$$z_k = x^* + \xi_k t_k \left(d + \frac{r(x^*, t_k)}{t_k} \right), \quad \xi_k \in (0, 1).$$

By (7.1), dividing by $t_k > 0$, we have

$$\frac{f(x_k) - f(x^*)}{t_k} = \nabla f(z_k)^T \left(d + \frac{r(x^*, t_k)}{t_k} \right). \quad (7.2)$$

Now, if $t_k \rightarrow 0$, we have also $r(x^*, t_k)/t_k \rightarrow 0$ and $z_k \rightarrow x^*$ and hence, as, by assumption, $\nabla f(x^*)^T d < 0$, by continuity of ∇f , for sufficiently large values of k , we must have

$$\nabla f(z_k)^T \left(d + \frac{r(x^*, t_k)}{t_k} \right) < 0$$

and hence, by (7.2) we obtain

$$f(x_k) - f(x^*) < 0,$$

which contradicts the hypothesis that x^* is a local minimizer. \square

7.4 Optimality Conditions for Problems with Equality and Inequality Constraints

We consider problems of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \quad h(x) = 0, \end{aligned} \quad (7.3)$$

where $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$ and $h : R^n \rightarrow R^p$ are continuously differentiable functions. The feasible set is now the set $S = \{x \in R^n : g(x) \leq 0, h(x) = 0\}$.

We start with a characterization of the tangent directions for S . Preliminarily we establish the following lemma.

Lemma 7.1 *Let $h : R^n \rightarrow R^p$ and let $x^* \in R^n$, such that $h(x^*) = 0$. We suppose that h is continuously differentiable on a neighborhood of x^* and that in this neighborhood the Jacobian matrix $\nabla h(x)^T$ has rank p . Then, given a non zero vector $d \in R^n$ there exists $\bar{t} > 0$ such that, for every $t \in (0, \bar{t}]$ we can find a point $x(t) \in R^n$ such that the following conditions hold.*

- (a) $\|x(t) - x^*\| \leq 2t\|d\|$;
- (b) $h(x(t)) = 0$;
- (c) the matrix $\nabla h(x(t))^T \nabla h(x(t))$ is nonsingular;
- (d) we have $P(x(t))(x(t) - x^* - td) = 0$, where

$$P(x) = I - \nabla h(x) \left[\nabla h(x)^T \nabla h(x) \right]^{-1} \nabla h(x)^T.$$

Proof Let $t > 0$ and define, for every integer $k > 0$, the problem

$$\min_x \Phi_k(x) \equiv \frac{1}{2t} \|x - x^* - td\|^2 + \frac{k}{2} \|h(x)\|^2. \quad (7.4)$$

It can be easily verified that the level set L_k of Φ_k , defined by

$$L_k = \{x \in R^n : \Phi_k(x) \leq \Phi_k(x^*)\}$$

is bounded. In fact, as $h(x^*) = 0$, for $x \in L_k$ we have

$$\frac{1}{2t} \|x - x^* - td\|^2 + \frac{k}{2} \|h(x)\|^2 \leq \frac{1}{2t} \|td\|^2, \quad (7.5)$$

which implies, in particular

$$\|x - x^*\| - \|td\| \leq \|x - x^* - td\| \leq \|td\|,$$

so that we have $\|x - x^*\| \leq 2\|td\|$ and the closed set L_k is also bounded and hence compact.

This in turn implies that there exists, for each k , a point $x_k \in L_k$ which minimizes Φ_k and satisfies $\|x_k - x^*\| \leq 2\|td\|$. Then there must exist a subsequence (which we relabel again $\{x_k\}$) converging to a point in L_k , which we denote by $x(t)$ for

evidencing the dependence on t . Thus we have

$$\|x(t) - x^*\| = \lim_{k \rightarrow \infty} \|x_k - x^*\| \leq 2\|td\|,$$

which establishes (a). Moreover, by (7.5) we can write

$$\frac{1}{2t} \|x_k - x^* - td\|^2 + \frac{k}{2} \|h(x_k)\|^2 \leq \frac{1}{2t} \|td\|^2. \quad (7.6)$$

This implies that the term $k\|h(x_k)\|^2$ must remain bounded for $k \rightarrow \infty$, which is possible only if $\|h(x_k)\| \rightarrow 0$ and hence, by continuity we have $h(x(t)) = 0$ and this proves (b).

Assertion (c) follows from the continuity of $\nabla h(x)$. In fact, by assertion (a), we can choose $t \leq \bar{t}$ sufficiently small to have that $\nabla h(x(t))$ has rank p . As x_k is an unconstrained minimizer of Φ_k in some open neighborhood of x^* where Φ_k is continuously differentiable, we must have $\nabla \Phi_k(x_k) = 0$, so that we can write

$$\nabla \Phi_k(x_k) = \frac{1}{t}(x_k - x^* - td) + k\nabla h(x_k)h(x_k) = 0, \quad (7.7)$$

where, for $t \leq \bar{t}$, the matrix $\nabla h(x_k)$ has rank p . Then, pre-multiplying both members by $\nabla h(x_k)^T$ and solving with respect to $kh(x_k)$, we obtain:

$$kh(x_k) = -\frac{1}{t} \left[\nabla h(x_k)^T \nabla h(x_k) \right]^{-1} \nabla h(x_k)^T (x_k - x^* - td),$$

and hence, by substituting this expression into the equality (7.7), we have that $P(x_k)(x_k - x^* - td) = 0$, where P is defined in (d). As x_k converges to $x(t)$, by continuity, assertion (d) is proved. \square

Now we can give the following characterization of tangent directions.

Proposition 7.4 Let $S = \{x \in R^n : h(x) = 0, g(x) \leq 0\}$ and let $x^* \in S$. Suppose that $h : R^n \rightarrow R^p$ and $g : R^n \rightarrow R^m$ are continuously differentiable in a neighborhood $B(x^*, \rho)$, with $\rho > 0$, of x^* and that the matrix $\nabla h(x^*)$ has rank p .

Suppose that $d \in R^n$ is a vector such that

$$\nabla h_i(x^*)^T d = 0, \quad i = 1, \dots, p, \quad \nabla g_i(x^*)^T d < 0, \quad i \in I_0(x^*), \quad (7.8)$$

where $I_0(x^*) = \{i : g_i(x^*) = 0\}$. Then d is a tangent direction for S at x^* .

Proof Let $d \in R^n$ be a vector satisfying (7.8). By Lemma 7.1 we can find $\bar{t} > 0$ such that, for all $t \in (0, \bar{t}]$, properties (a)(b) (c) and (d) of the lemma are satisfied. We can also assume that \bar{t} is sufficiently small to have that, for all x, t such that $t \in (0, \bar{t}]$ and $\|x - x^*\| \leq 2t\|d\|$, x belongs to $B(x^*, \rho)$ and also all the inequality constraints negative in x^* are negative at x , that is:

$$g_i(x) < 0, \quad \text{for all } i \notin I_0(x^*) \text{ and for all } x \text{ such that } \|x - x^*\| \leq 2t\|d\|. \quad (7.9)$$

Let now $\{t_k\}$ be a sequence of positive numbers $t_k \in (0, \bar{t}]$ such that $t_k \rightarrow 0$. By Lemma 7.1 and (7.9) we can determine a sequence of points $x(t_k)$ such that

$$\|x(t_k) - x^*\| \leq 2t_k\|d\| \quad (7.10)$$

$$h(x(t_k)) = 0 \quad (7.11)$$

$$P(x(t_k))(x(t_k) - x^* - t_k d) = 0 \quad (7.12)$$

$$g_i(x(t_k)) < 0 \quad i \notin I(x^*) \quad (7.13)$$

where $P(x(t_k))$ has rank p . By (7.10) the sequence of vectors $(x(t_k) - x^*)/t_k$ is bounded and hence we extract a subsequence of $\{x(t_k)\}$, which we relabel $\{x(t_k)\}$, such that

$$\lim_{k \rightarrow \infty} (x(t_k) - x^*)/t_k = \bar{d}, \quad (7.14)$$

where \bar{d} is a point in R^n . Moreover, again by (7.10), we have that $x(t_k) \rightarrow x^*$. Using the Theorem of the mean we can write

$$\frac{1}{t_k} h_i(x(t_k)) = \frac{1}{t_k} h_i(x^*) + \frac{1}{t_k} \nabla h_i(z(i, t_k))^T (x(t_k) - x^*), \quad i = 1, \dots, p,$$

where

$$z(i, t_k) = x^* + \xi_i^k (x(t_k) - x^*),$$

with $\xi_i^k \in (0, 1)$. As $h_i(x(t_k)) = 0$ and $h_i(x^*) = 0$, taking limits for $k \rightarrow \infty$ and taking into account the limits stated above, we obtain $\nabla h_i(x^*) = 0$ for $i = 1, \dots, p$ and hence we can write

$$\nabla h(x^*)^T \bar{d} = 0. \quad (7.15)$$

Moreover, dividing both members of (7.12) by t_k and taking limits, we have that $P(x^*)(d - \bar{d}) = 0$, which implies, by definition of P

$$(\bar{d} - d) - \nabla h(x^*) \left[\nabla h(x^*)^T \nabla h(x^*) \right]^{-1} \nabla h(x^*)^T (\bar{d} - d) = 0. \quad (7.16)$$

Using (7.8) and (7.15) we obtain

$$\bar{d} = d.$$

Then, by (7.14) we can write

$$\lim_{k \rightarrow \infty} \frac{(x(t_k) - x^*)}{t_k} = d. \quad (7.17)$$

Now consider the inequality constraints that are active at x^* . Using the Theorem of the mean, we have

$$\begin{aligned} \frac{1}{t_k} g_i(x(t_k)) &= \frac{1}{t_k} g_i(x^*) + \frac{1}{t_k} \nabla g_i(w(i, t_k))^T (x(t_k) - x^*) \\ &= \nabla g_i(w(i, t_k))^T \frac{(x(t_k) - x^*)}{t_k}, \quad i \in I_0(x^*), \end{aligned} \quad (7.18)$$

where $w(i, t_k) = x^* + \zeta_i^k (x(t_k) - x^*)$ with $\zeta_i^k \in (0, 1)$. Taking limits for $k \rightarrow \infty$ and recalling (7.17) and the assumptions on d we have

$$\lim_{k \rightarrow \infty} \nabla g_i(w(i, t_k))^T \frac{(x(t_k) - x^*)}{t_k} = \nabla g_i(x^*)^T d < 0, \quad i \in I_0(x^*).$$

Then, by continuity, it follows from (7.18) that for sufficiently small values of \bar{t} we have

$$g_i(x(t_k)) < 0, \quad i \in I_0(x^*), \quad t_k \in (0, \bar{t}]. \quad (7.19)$$

Thus we can conclude that, for a suitable choice of \bar{t} , we can construct a sequence $\{x(t_k)\}$, such that the direction d satisfies the conditions given in Definition 7.1 and hence it is a tangent direction for S at x^* . \square

Making use of the characterization given in the preceding proposition, we can give a necessary optimality condition.

Proposition 7.5 *Let $S = \{x \in R^n : h(x) = 0, g(x) \leq 0\}$ and let $x^* \in S$ be a local minimizer of f . Suppose that f, g, h are continuously differentiable in a neighborhood of x^* and that the matrix $\nabla h(x^*)$ has rank p .*

Then the following system has no solution:

$$\nabla f(x^*)^T d < 0, \quad \nabla h(x^*)^T d = 0, \quad \nabla g_{I_0(x^*)}(x^*)^T d < 0, \quad (7.20)$$

where $\nabla g_{I_0(x^)}$ is the matrix with columns $\nabla g_i(x^*)$ for $i \in I_0(x^*)$.*

Proof If system (7.20) has a solution, it follows from Proposition 7.4 that d is a tangent direction, which is also a descent direction and this contradicts, as we already know, the assumption that x^* is a local minimizer. \square

7.5 Fritz John Conditions Established with Motzkin Theorem

Fritz John conditions can be obtained easily from the Proposition 7.5 by employing theorems of the alternative. In fact, we can give a new derivation of FJ conditions, (which we restate for convenience of the reader) as shown in the following proposition.

Proposition 7.6 (Fritz John Necessary Conditions) *Let x^* be a local minimum point of Problem (7.3) and suppose that the functions f, g, h are continuously differentiable in an open neighborhood of x^* .*

Then there exist multipliers $\lambda_0^ \in R$, $\lambda^* \in R^m$, $\mu^* \in R^p$ such that the following conditions hold*

$$\begin{aligned} \lambda_0^* \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^p \mu_i^* \nabla h_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0, \quad i = 1, \dots, m \\ (\lambda_0^*, \lambda^*) &\geq 0, \quad (\lambda_0^*, \lambda^*, \mu^*) \neq 0, \\ g(x^*) &\leq 0, \quad h(x^*) = 0. \end{aligned} \tag{7.21}$$

Proof We observe first that, if $\nabla h(x^*)$ has some linearly dependent columns, there must exist a vector $v \in R^p$, $v \neq 0$ such that

$$\sum_1^p v_i \nabla h_i(x^*) = 0.$$

In this case the FJ conditions trivially hold by taking

$$\lambda_0^* = 0, \lambda^* = 0, \mu^* = v.$$

Thus, we can assume that $\nabla h(x^*)$ has rank p . By Proposition 7.5 the following system has no solution

$$\begin{pmatrix} \nabla f(x^*)^T \\ \nabla g_{I_0(x^*)}(x^*)^T \end{pmatrix} d < 0, \quad \nabla h(x^*)^T d = 0.$$

Then, by Motzkin Theorem there must exist a solution $(\lambda_0^*, y^*, \mu^*)$ of the alternative system

$$\nabla f(x^*)\lambda_0^* + \nabla g_{I_0(x^*)}(x^*)y^* + \nabla h(x^*)\mu^* = 0,$$

where $\lambda_0^* \geq 0$, $y^* \geq 0$ and $(\lambda_0^*, y^*) \neq 0$. Then, letting

$$\lambda_i^* = \begin{cases} 0 & \text{if } i \notin I_0(x^*) \\ y_i^* & \text{if } i \in I_0(x^*) \end{cases}$$

we can write

$$\nabla f(x^*)\lambda_0^* + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* = 0,$$

and hence, by definition of λ_i^* we have that

$$\lambda_i^* g_i(x^*) = 0, \quad i = 1 \dots, m.$$

We can also easily verify that $(\lambda_0^*, \lambda^*, \mu^*) \neq 0$. In fact, as already observed, if $\text{rank}(\nabla h(x^*)) < p$ we can satisfy the assertions by choosing

$$\lambda_0^* = 0, \lambda^* = 0, \mu^* = \nu.$$

with $\nu \neq 0$. If $\nabla h(x^*)$ has rank p , it follows from Motzkin theorem and our definition of λ^* that $(\lambda_0^*, \lambda^*) \neq 0$. \square

7.5.1 KKT Conditions from FJ Conditions

Starting from FJ conditions, we can impose constraints qualifications under which also the KKT conditions are satisfied.

First we introduce the following condition that we call Kuhn-Tucker (KT) constraint qualification.

Definition 7.1 (KT Constraint Qualification (KTCQ)) Let $\hat{x} \in S$ and suppose that the functions f, g, h are continuously differentiable in an open neighborhood of \hat{x} . We say that the Kuhn-Tucker constraint qualification holds at \hat{x} if every non zero direction $d \in R^n$ such that

$$\nabla g_{I_0(\hat{x})}(\hat{x})^T d \leq 0, \quad \nabla h(\hat{x})^T d = 0$$

is a tangent direction for S at \hat{x} . \square

Then we can state the following proposition.

Proposition 7.7 (KKT Optimality Conditions Under KTCQ) Let $x^* \in S$ be a local minimum point and suppose that the functions f, g, h are continuously differentiable in an open neighborhood of x^* .

Suppose further that the KTCQ holds at x^* . Then there exist $\lambda^* \in R^m$ and $\mu^* \in R^p$ such that the following KKT conditions hold.

$$\begin{aligned} \nabla f(x^*) + \nabla g(x^*)\lambda + \nabla h(x^*)\mu^* &= 0 \\ g(x^*) &\leq 0, \quad h(x^*) = 0 \\ \lambda^{*T} g(x^*) &= 0, \quad \lambda^* \geq 0. \end{aligned} \tag{7.22}$$

Proof By the KTCQ we have that the system

$$\nabla f(x^*)^T d < 0, \quad \nabla g_{I(x^*)}(x^*)^T d \leq 0, \quad \nabla h(x^*)^T d = 0$$

cannot admit a solution, for otherwise, d would be a descent direction which is also a tangent direction at x^* , which is impossible by Proposition 7.3. Then, by Motzkin Theorem, this implies that the alternative system

$$\lambda_0^* \nabla f(x^*) + \nabla g_{I(x^*)}(x^*) y_{I(x^*)}^* + \nabla h(x^*) \mu^* = 0$$

must admit a solution with $\lambda_0^* \geq 0$, $\lambda_0^* \neq 0$, and $y_{I(x^*)}^* \geq 0$. As $\lambda_0^* \in R$ we can obviously take $\lambda_0^* > 0$. Letting

$$\lambda_i^* = \begin{cases} 0 & i \notin I(x^*) \\ y_i^* & i \in I(x^*) \end{cases}$$

the FJ conditions hold with $\lambda_0^* > 0$ and hence the KKT conditions hold, dividing by $\lambda_0^* > 0$, and redefining the multipliers. \square

An important special case is when the functions $g_i, i \in I_0(x^*)$ are concave at x^* in a neighborhood of x^* and the functions h_i are affine. We can easily see that in this case the KTCQ is satisfied, since there exists a feasible direction d at x^* . In fact, suppose that there exists a direction d such that

$$\nabla g_{I_0(x^*)}(x^*)^T d \leq 0, \quad \nabla h(x^*)^T d = 0.$$

Then, letting $t \in (0, \bar{t}]$, for $\bar{t} > 0$ sufficiently small, we have that, for every $i \in I_0(x^*)$, the points $x^* + td$ remain in the neighborhood where g_i is concave, so that

$$g_i(x^* + td) \leq g_i(x^*) + t \nabla g_i(x^*)^T d \leq 0.$$

Similarly, for every $i = 1, \dots, p$ we have

$$h_i(x^* + td) = h_i(x^*) + t \nabla h_i(x^*)^T d = 0.$$

This proves that d is a feasible direction and that the KTCQ is satisfied.

Then we can assert that when the active constraints at x^* are concave in a neighborhood of this point, the KKT conditions must be satisfied.

It is easily seen that all the remaining constraint qualifications stated in Chap. 5 can be used here for deriving the KKT conditions.

7.6 Appendix

In this appendix we start from some elementary remarks on system of linear inequalities and we give some basic results on theorems of the alternative. It will be also shown that these results yield directly the optimality conditions for Linear Programming, using only an algebraic approach.

7.6.1 Linear Inequalities and Theorems of the Alternative

The study of linear inequalities has a major role in mathematical programming, both because in many problems we have a feasible set defined by linear inequalities, and also because we can derive, in general, optimality conditions through a *local linearization* of the objective function and of the constraints.

When both the objective and the constraints are linear, that is in linear programming, we can show that the optimization problem is equivalent to a system of linear inequalities.

We know that, in general, it is neither possible to express analytically the solution of a system of linear inequalities, nor to give analytical conditions on the coefficients of the system of inequalities to establish existence or non existence of solutions. It is possible, however, to prove the equivalence between establishing the *non existence of solutions* of a given system and the *existence* of solution of another system known as the *alternative system*.

Given two systems (I) and (II), an alternative result consists in proving that *system (I) has a solution, if and only if system (II) has no solution.*

If we denote by (I) (or (II)) the assertion that (I) (or (II)) has a solution and by $\neg(I)$ ($\neg(II)$) the negation of an assertion, that is the assertion that (I) (or (II)) has no solution, the alternative can be expressed in the form

$$(I) \Leftrightarrow \neg(II).$$

This can be established, for instance, by proving that

- (a) $(I) \rightarrow \neg(II)$
- (b) $\neg(II) \rightarrow (I)$.

The same result can be proved in different ways by employing the *transposition rule* of propositional logic, which consists in establishing that the assertion

“the truth of A implies the truth of B”

is “equivalent”, in a logical proof, to the assertion

“the truth of not-B implies the truth of not-A”.

This rule can be expressed formally as:

$$(A \rightarrow B) \Leftrightarrow (\neg B \rightarrow \neg A).$$

Thus, in our case, we can prove, for instance, a result of alternative, also by proving the implications

- (a) $(I) \rightarrow \neg(II)$
- (c) $\neg(I) \rightarrow (II)$,

where (c) can replace (b) given above, by the transposition rule, noting that

$$\neg(\neg(II)) \rightarrow (II).$$

Before establishing the main results of the alternative we report some preliminary material on equivalent transformations of linear systems.

7.6.1.1 Equivalent Transformations

In the sequel we will often need to transform a given system (S) into an equivalent system (S') in the sense that *system (S) has a solution if and only if system (S') has a solution*. We will indicate by A a given $m \times n$ real matrix and by x an element of R^n . An obvious equivalence is given below.

(a) Transformation of Equations into Inequalities

The system (S): $Ax = b$ is equivalent to (S'): $\begin{pmatrix} A \\ -A \end{pmatrix} x \geq \begin{pmatrix} b \\ -b \end{pmatrix}$. \square

It can be easily shown that also the next equivalence holds.

(b) Homogenization

The system (S): $Ax \geq b$ is equivalent to the homogeneous system

$$(S'): \quad (A - b) \begin{pmatrix} x \\ \zeta \end{pmatrix} \geq 0, \quad (0^T 1) \begin{pmatrix} x \\ \zeta \end{pmatrix} > 0.$$

where $\zeta \in R$ and 0^T is the null matrix $1 \times n$. \square

In fact, we can note that if x solves (S) then $(x, 1)$ solves (S'). Conversely, if (x, ζ) solves (S'), we have $\zeta > 0$ and hence the point

$$\bar{x} = x/\zeta$$

is a solution to (S).

By following a similar reasoning, an equality system of the form $Ax = b$, will be equivalent to

$$(A - b) \begin{pmatrix} x \\ \zeta \end{pmatrix} = 0, \quad (0^T 1) \begin{pmatrix} x \\ \zeta \end{pmatrix} > 0.$$

The next equivalence shows a system, which imposes that a vector is non-zero in an homogeneous system

(c) Non Zero Vector in an Homogeneous System

The system (S)

$$Ax \geq 0, \quad x \geq 0 \quad x \neq 0$$

is equivalent to (S')

$$Ax \geq 0 \quad x \geq 0 \quad e^T x = 1,$$

where $e \in R^n$ is the vector with components $e_i = 1, i = 1, \dots, n$. \square

To justify the preceding assertion, we can observe that if x solves (S), as $x \neq 0$, we have

$$e^T x = \sum_{j=1}^n x_j > 0$$

and hence the vector $\bar{x} = x/e^T x$ is a solution to (S'). Conversely, a solution to (S') must necessarily satisfy $x \neq 0$ and hence will be a solution to (S).

The next transformation will be frequently used.

(d) Transformation of Strict (Homogeneous) Inequalities into Closed Inequalities

The system (S): $Ax > 0$ is equivalent to (S'): $Ax \geq e$, where $e \in R^n$ is the vector with components $e_i = 1, i = 1, \dots, n$. \square

As $e > 0$, a solution x to (S') is obviously a solution to (S). Assume now that x is a solution to (S) and set $Ax = y > 0$.

Let now γ be the number $\gamma = \min\{y_i, i = 1, \dots, m\}$. Then $y_i/\gamma \geq 1$ and hence

$$\frac{y}{\gamma} \geq e.$$

Then, setting $\bar{x} = x/\gamma$ and recalling that $y = Ax$, we obtain

$$A\bar{x} = \frac{1}{\gamma}Ax = \frac{y}{\gamma} \geq e,$$

so that \bar{x} solves (S').

If system (S) is not homogeneous, of the form $Ax > b$, we must first transform the system into an homogeneous system, in the form

$$\begin{pmatrix} A & -b \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x \\ \zeta \end{pmatrix} > 0,$$

so that we have the following transformation.

(e) Transformation of Strict Inequalities into Closed Inequalities

The system (S)

$$Ax > b$$

is equivalent to (S')

$$\begin{pmatrix} A & -b \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} x \\ \zeta \end{pmatrix} \geq \begin{pmatrix} e \\ 1 \end{pmatrix},$$

where $e \in R^n$ is the vector with components $e_i = 1, i = 1, \dots, n$.

□

7.6.1.2 A Theorem of the Alternative for Linear Equations

We state here a theorem of the alternative for linear equations, which can be viewed as a special case of the Fredholm alternative theorem on function spaces.

Theorem 7.1 (Fredholm Theorem) *The system*

$$Ax = 0, \quad c^T x = 1, \tag{I}$$

with A matrix $m \times n$ and $c \in R^m$, has solution if and only if the system

$$A^T \lambda = c \tag{II}$$

with $\lambda \in R^m$ has no solution.

Proof First we show that (I) \rightarrow \neg (II). Let \bar{x} be a solution of (I) and assume, by contradiction, that there exists a solution $\bar{\lambda}$ of (II). Then, we can write:

$$1 = c^T \bar{x} = (A^T \bar{\lambda})^T \bar{x} = \bar{\lambda}^T A \bar{x} = \bar{\lambda}^T (A \bar{x}) = 0,$$

which is a contradiction.

Now we prove that \neg (I) \rightarrow (II). System (I) can be rewritten in the form:

$$\begin{pmatrix} A \\ c^T \end{pmatrix} x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

From the theory of linear equations we know that this system has no solution if and only if

$$\text{rank} \begin{pmatrix} A \\ c^T \end{pmatrix} \neq \text{rank} \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix}.$$

Therefore, we must have necessarily that

$$\text{rank} \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix} = \text{rank} \begin{pmatrix} A \\ c^T \end{pmatrix} + 1. \quad (7.23)$$

On the other hand, the last row ($c^T 1$) is linearly independent on the rows of ($A 0$) and hence :

$$\text{rank} \begin{pmatrix} A & 0 \\ c^T & 1 \end{pmatrix} = \text{rank} (A 0) + 1 = \text{rank}(A) + 1. \quad (7.24)$$

From (7.23) and (7.24) we obtain

$$\text{rank} (A^T c) = \text{rank} \begin{pmatrix} A \\ c^T \end{pmatrix} = \text{rank}(A) = \text{rank}(A^T)$$

and this implies that the system $A^T \lambda = c$ has a solution. \square

7.6.1.3 Implications of an Inequality System

We introduce the following definition.

Definition 7.2 (Implication) Let A be an $m \times n$ real matrix, and let $b \in R^m$ and $c \in R^n$. We say that the inequality $c^T x \leq d$ is an *implication* of the system $Ax \leq b$ if every solution of $Ax \leq b$ satisfies $c^T x \leq d$, or equivalently, if the system

$$Ax \leq b \quad c^T x > d$$

has no solution. \square

It easily seen, from a geometrical point of view, that $c^T x \leq d$ is an implication of $Ax \leq b$ if the polyhedron defined by $Ax \leq b$ is contained in the closed halfspace defined by $c^T x \leq d$.

An immediate consequence of the above definition is the following proposition

Proposition 7.8 Every nonnegative combination of the inequalities of the system $Ax \leq c$, where $A(m \times n)$, $x \in R^n$, $c \in R^m$, is an implication of the system.

Proof Let $\lambda \in R^m$ be a vector such that $\lambda \geq 0$. Consider the inequality $\lambda^T Ax \leq \lambda^T c$ obtained as nonnegative combination of the inequalities of the system $Ax \leq c$. Then as $\lambda \geq 0$, if \bar{x} satisfies $A\bar{x} \leq c$, we have obviously $\lambda^T A\bar{x} \leq \lambda^T c$. \square

Now we establish the following lemmas.

Lemma 7.2 Let A be a matrix $m \times n$ and $c \in R^n$. If $c^T x \leq 0$ is an implication of the system $Ax \leq 0$, then we have $c = A^T \lambda$ with $\lambda \in R^m$.

Proof If $c^T x \leq 0$ is an implication of the system $Ax \leq 0$ cannot exist, in particular, an $x \in R^n$ such that $Ax = 0$ and $c^T x = 1$. Thus system (I) in Theorem 7.1 has no solution and hence our thesis follows directly from Theorem 7.1. \square

Lemma 7.3 Let $c^T x \leq 0$ be an implication of $Ax \leq 0$, and assume also that

$$c = A^T \lambda \quad \text{with} \quad \lambda_1 \dots \lambda_{m-1} \geq 0 \quad \text{and} \quad \lambda_m \leq 0. \quad (7.25)$$

Then $c^T x \leq 0$ is also an implication of the system obtained from $Ax \leq 0$ by eliminating the m -th inequality.

Proof By contradiction, let $a_i \in R^m$ be the rows of A for $i = 1, \dots, m$ and suppose there exists \bar{x} such that

$$a_i^T \bar{x} \leq 0, \quad i = 1, \dots, m-1 \quad c^T \bar{x} > 0. \quad (7.26)$$

Recalling the definition of c in (7.25), we can write:

$$0 < c^T \bar{x} = (A^T \lambda)^T \bar{x} = \lambda^T (A\bar{x}) = \sum_{i=1}^{m-1} \lambda_i a_i^T \bar{x} + \lambda_m a_m^T \bar{x},$$

where $a_i^T \bar{x} \leq 0$ for $i = 1, \dots, m$, $\lambda_i \geq 0$ for $i = 1, \dots, m-1$ and $\lambda_m \leq 0$. This is possible only if $a_m^T \bar{x} \leq 0$ and hence, by (7.26), we have:

$$a_i^T \bar{x} \leq 0, \quad i = 1, \dots, m, \quad c^T \bar{x} > 0,$$

which contradicts the assumption that $c^T x \leq 0$ is an implication of $Ax \leq 0$. \square

7.6.1.4 Farkas Lemma

Farkas Lemma (known also as Minkowski Theorem) is one of the best known theorems of the alternative and it can be employed for deriving optimality conditions in Linear Programming and also, under suitable assumptions, in Nonlinear Programming. It can be stated, for instance, in the following form.

Theorem 7.2 (Farkas Lemma) *The system*

$$A^T \lambda = c, \quad \lambda \geq 0, \quad (\text{I})$$

with A $m \times n$ real matrix, and $c \in R^n$ has a solution $\lambda \in R^m$, if and only if the system

$$Ax \leq 0, \quad c^T x > 0 \quad (\text{II})$$

with $x \in R^n$ has no solution, that is, if and only if $c^T x \leq 0$ is an implication of $Ax \leq 0$.

Proof First we show that (II) \rightarrow $\neg(\text{I})$. Let \bar{x} be a solution of system (II) and assume, by contradiction, that there exists a solution $\bar{\lambda}$ of system (I); then:

$$A\bar{x} \leq 0 \rightarrow \bar{\lambda}^T A\bar{x} \leq 0 \rightarrow c^T \bar{x} \leq 0,$$

which contradicts the assumption that \bar{x} is a solution to (II).

Now we show that $\neg(\text{II})\rightarrow(\text{I})$. If (II) has no solution the inequality $c^T x \leq 0$ is an implication of $Ax \leq 0$, and hence, by Lemma 7.2, there exists $\lambda \in R^m$ such that

$$c = \sum_{i=1}^m \lambda_i a_i. \quad (7.27)$$

We must prove that there exists a representation of the form (7.27) such that $\lambda \geq 0$. The proof is by induction on the number of inequalities, that is, on the number of rows of the matrix A .

First we show that the assertion is true for $m = 1$, and hence we suppose that $c^T x \leq 0$ is an implication of $a_1^T x \leq 0$. By (7.27) we have $c = \lambda a_1$ with $\lambda \in R$. If $c = 0$, the assertion is true by assuming $\lambda = 0$. If $c \neq 0$ we have also $a_1 \neq 0$ and we can consider the vector $\bar{x} = -a_1 \neq 0$. It follows that $a_1^T \bar{x} = -\|a_1\|^2 < 0$ and hence, by assumption, we have $c^T \bar{x} \leq 0$ (as $c^T x \leq 0$ must be an implication of $a_1^T x \leq 0$). Then we obtain

$$c^T \bar{x} = (\lambda a_1)^T (-a_1) = -\lambda \|a_1\|^2 \leq 0,$$

which implies $\lambda \geq 0$.

Now assume that the assertion is true for a matrix with $m - 1$ rows (*induction hypothesis*) and we show that the same is true for a matrix with m rows.

Let us choose a vector $\bar{\lambda}$ such that this vector has the maximum number s of nonnegative components among the vectors that satisfy (7.27). We must show that $s = m$. Reasoning by contradiction, suppose that $s < m$ and, after a reordering, if needed, we can write $\bar{\lambda}_1, \dots, \bar{\lambda}_s \geq 0$ and $\bar{\lambda}_{s+1}, \dots, \bar{\lambda}_m < 0$.

If we define the vector

$$d = \sum_{i=1}^s \bar{\lambda}_i a_i + \bar{\lambda}_m a_m \quad (7.28)$$

we can write, by (7.27):

$$c = d + \sum_{i=s+1}^{m-1} \bar{\lambda}_i a_i. \quad (7.29)$$

Multiplying by a vector x such that $a_i^T x \leq 0$ for $i = 1, \dots, m$, and recalling that, by assumption, $c^T x \leq 0$ is an implication of $a_i^T x \leq 0$ with $i = 1, \dots, m$, from (7.29) it follows that

$$0 \geq c^T x = d^T x + \sum_{i=s+1}^{m-1} \bar{\lambda}_i a_i^T x \quad \text{for all } x : a_i^T x \leq 0 \quad i = 1, \dots, m.$$

As $\bar{\lambda}_{s+1}, \dots, \bar{\lambda}_{m-1} < 0$ we must have necessarily:

$$d^T x \leq 0 \quad \text{for all } x : a_i^T x \leq 0 \quad i = 1, \dots, m,$$

so that $d^T x \leq 0$ is an implication of the inequalities $a_i^T x \leq 0$ for $i = 1, \dots, m$. On the other hand, by definition (7.28), d is obtained as linear combination of

the vectors a_i with nonnegative coefficients, except $\bar{\lambda}_m$. Thus, by Lemma 7.3 the inequality $d^T x \leq 0$ is also an implication of $a_i^T x \leq 0$ for $i = 1, \dots, m - 1$.

Then, from the induction hypothesis, it follows that d is a nonnegative combination of the $m - 1$ rows a_i , that is, there exists $\mu \in R^{m-1}$ such that

$$\sum_{i=1}^{m-1} \mu_i a_i = d, \quad \mu_i \geq 0, \quad i = 1, \dots, m - 1.$$

Then, we can write

$$\begin{aligned} c = d + \sum_{i=s+1}^{m-1} \bar{\lambda}_i a_i &= \sum_{i=1}^{m-1} \mu_i a_i + \sum_{i=s+1}^{m-1} \bar{\lambda}_i a_i \\ &= \sum_{i=1}^s \mu_i a_i + \sum_{i=s+1}^{m-1} (\bar{\lambda}_i + \mu_i) a_i + 0 \cdot a_m \end{aligned}$$

It follows that c can be expressed as a linear combination of the columns of A with $s + 1$ non negative coefficients $\mu_1, \dots, \mu_s, 0$ (the m -th column is multiplied by 0). This yields a contradiction, as we assumed that s is the maximum number of nonnegative components in the definition of c . \square

We can give an immediate geometric interpretation of Farkas Lemma, recalling that the angle between two vectors is acute, right, or obtuse if the scalar product of the vectors is positive, zero or negative.

In fact, suppose that the rows (transposed) of A , that is the vectors $a_i \in R^n$, for $i = 1, \dots, m$ and the vector $c \in R^n$ are given. If system (I) has a solution this implies that there exists a vector y such that

$$\begin{aligned} c^T y < 0 &\quad (y \text{ makes an obtuse angle with } c); \\ a_i^T y \geq 0, \quad i = 1, \dots, q &\quad (y \text{ makes non obtuse angles with all vectors } a_i). \end{aligned}$$

Farkas Lemma states that this can happen if and only if c does not belong to the convex cone generated by vectors a_i , that is if and only if do not exist vectors u_i such that

$$c = \sum_{i=1}^q u_i a_i, \quad u_i \geq 0, \quad i = 1, \dots, q,$$

which is equivalent to system (II).

Farkas Lemma can be rewritten as an existence condition for a system in *standard form*. We reformulate the problem by employing the notation usually employed for writing the system in standard form, that is, as a system of linear equations with nonnegative variables.

Theorem 7.3 (Farkas Lemma) Let A be a $(m \times n)$ real matrix, $x \in R^n$ and $b \in R^m$.

The system

$$Ax = b \quad x \geq 0, \quad (\text{I})$$

has a solution $x \in R^n$, if and only if the system

$$A^T u \leq 0 \quad b^T u > 0 \quad (\text{II})$$

with $u \in R^m$, has no solution.

7.6.1.5 Other Alternative Theorems

From Farkas Lemma we can derive many other alternative theorems using equivalent transformations of system (I) or system (II) in Farkas Lemma. One of the most general results is Motzkin Theorem.

Theorem 7.4 (Motzkin Theorem) Let $B(m \times n)$, $C(p \times n)$, $D(q \times n)$ given real matrices with $B \neq 0$. The system

$$(\text{I}) \quad Bx > 0, \quad Cx \geq 0, \quad Dx = 0$$

has solution $x \in R^n$ if and only if the system

$$(\text{II}) \quad B^T y + C^T z + D^T u = 0, \quad y \geq 0, \quad y \neq 0, \quad z \geq 0,$$

has no solution with $y \in R^m$, $z \in R^p$, $u \in R^q$.

Proof The proof can be derived from Farkas Lemma. By employing the equivalent transformations introduced before, and letting $e \in R^m$ be the unit vector, we can put system (II) in the form

$$B^T y + C^T z + D^T(u^+ - u^-) = 0, \quad e^T y = 1, \quad y \geq 0, \quad z \geq 0, \quad u^+ \geq 0, \quad u^- \geq 0$$

which corresponds to the system in standard form

$$\begin{pmatrix} B^T & C^T & D^T & -D^T \\ e^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y \\ z \\ u^+ \\ u^- \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} y \\ z \\ u^+ \\ u^- \end{pmatrix} \geq 0.$$

Table 7.1 Theorems of the alternative. System (I) has a solution if and only if system (II) has no solution

System (I)	System (II)	Name
$Bx > 0, Cx \geq 0$	$B^T y + C^T z = 0, y \geq 0, y \neq 0, z \geq 0$	(Motzkin)
$Bx > 0, Cx \geq 0, Dx = 0$	$B^T y + C^T z + D^T u = 0, y \geq 0, y \neq 0, z \geq 0$	(Motzkin)
$Ax > 0$	$A^T u = 0, u \geq 0, u \neq 0$	(Gordan)
$Ax > 0, x > 0$	$A^T u \leq 0, u \geq 0, u \neq 0$	(Ville)
$Ax \geq b$	$A^T u = 0, u \geq 0, b^T u > 0$	(Gale)
$Ax = b, x \geq 0$	$A^T u \leq 0, b^T u > 0$	(Farkas)
$Ax = b$	$A^T u = 0, b^T u > 0$	(Gale, Fredholm)
$Ax \geq b, x \geq 0$	$A^T u \leq 0, u \geq 0, b^T u > 0$	
$c^T x < \beta, Ax \geq b$	$A^T u = 0, u \geq 0, b^T u > 0$ and:	(Non homog. Farkas)
$Ax \geq b, Cx = d$	$A^T u + C^T v = 0, u \geq 0, b^T u + d^T v > 0$	
$Ax \geq b, Cx = d, x \geq 0$	$A^T u + C^T v \leq 0, u \geq 0, b^T u + d^T v > 0$	

By identifying this system with system (I) in Farkas Lemma we have that the system has a solution if and only if the alternative system has no solution.

The alternative system has the form

$$Bv + e\xi \leq 0, Cv \leq 0, Dv \leq 0, -Dv \leq 0, \xi > 0, v \in R^n$$

and hence, letting $x = -v$, it easily seen that the equivalent system

$$Bx > 0, Cx \geq 0, Dx = 0,$$

has no solution. This completes the proof. \square

The main theorems of the alternative and some simple consequences are displayed in Table 7.1. We refer to [179] for additional results on alternative systems.

We show, as example, how we can derive from Farkas Lemma the Gale Theorem reported below, which gives a necessary and sufficient existence condition, in terms of alternative, for a system of linear inequalities.

Theorem 7.5 (Gale Theorem) Let A an $m \times n$ real matrix and $b \in R^m$. The system

$$Ax \geq b, \text{ with } x \in R^n \quad (\text{I})$$

has solution if and only if the system

$$b^T u > 0, A^T u = 0, u \geq 0 \text{ with } u \in R^m \quad (\text{II})$$

has no solution.

Proof Let $x = x^+ - x^-$, with $x^+ \geq 0$, $x^- \geq 0$ and let $z \geq 0$. We can easily verify that (I) has solution if and only if the equivalent system in standard form

$$(I') \quad (A - A - I) \begin{pmatrix} x^+ \\ x^- \\ z \end{pmatrix} = b, \quad \begin{pmatrix} x^+ \\ x^- \\ z \end{pmatrix} \geq 0,$$

has a solution. By Farkas Lemma we have that (I') (and hence (I)) has a solution if and only if the system

$$(II') \quad b^T u > 0, \quad \begin{pmatrix} A^T \\ -A^T \\ -I \end{pmatrix} u \leq 0,$$

which is equivalent to our system (II), has no solution. \square

7.6.2 Non Homogeneous Farkas Lemma and Optimality for LP

We will establish the optimality conditions for Linear Programming by employing a non homogeneous version of Farkas Lemma considered below.

7.6.2.1 Non Homogeneous Farkas Lemma

We can easily derive from Motzkin theorem the following result.

Theorem 7.6 (Non Homogeneous Farkas Lemma) *Let $A(m \times n)$, $c \in R^n$, $b \in R^m$ and $\beta \in R$. . The system*

$$c^T x < \beta, \quad Ax \geq b, \tag{I}$$

has solution $x \in R^n$ if and only if neither of the two following systems has a solution in R^m

$$b^T u \geq \beta, \quad A^T u = c, \quad u \geq 0, \tag{II'}$$

$$b^T u > 0, \quad A^T u = 0, \quad u \geq 0 \tag{II''}$$

Proof It can be easily verified that (I) has solution if and only if the following homogeneous system has a solution $x \in R^n, \xi \in R$:

$$(I') \quad -c^T x + \beta \xi > 0, \quad \xi > 0, \quad Ax - b\xi \geq 0$$

We can rewrite this system into the following form.

$$(I') \quad \begin{pmatrix} -c^T & \beta \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ \xi \end{pmatrix} > 0, \quad (A - b) \begin{pmatrix} x \\ \xi \end{pmatrix} \geq 0.$$

Then, letting

$$B = \begin{pmatrix} -c^T & \beta \\ 0 & 1 \end{pmatrix}, \quad C = (A - b), \quad D = 0$$

and using Motzkin Theorem, we have that system (I') (and hence system (I)) has a solution if and only if has no solution the system

$$\begin{pmatrix} -c & 0 \\ \beta & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} A^T \\ -b^T \end{pmatrix} u = 0, \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq 0, \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \neq 0, \quad u \geq 0,$$

where $y_1 \in R, y_2 \in R, u \in R^m$.

This is equivalent to say that (I) has a solution if and only if has no solution the system

$$-cy_1 + A^T u = 0, \quad \beta y_1 + y_2 - b^T u = 0, \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq 0, \quad \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \neq 0, \quad u \geq 0. \quad (7.30)$$

Now we can distinguish the two cases:

- (a) $y_1 > 0$ and $y_2 \geq 0$;
- (b) $y_1 = 0$ and $y_2 > 0$.

In the first case, dividing both equations by $y_1 > 0$, we get the equivalent system

$$(II') \quad b^T u \geq \beta, \quad A^T u = c, \quad u \geq 0.$$

In the second case, we have:

$$(II'') \quad b^T u > 0, \quad A^T u = 0, \quad u \geq 0.$$

This proves our assertion. In fact, if (I) has solution, system (7.30) has no solution and hence neither of the two systems (II') and (II'') can have solution. Conversely, if (I) has no solution then system (7.30) has a solution and hence at least one of the two systems (II') o (II'') must admit a solution. \square

7.6.2.2 Optimality Conditions for LP

Consider now a linear programming problem of the form

$$\begin{aligned} \min c^T x \\ Ax \geq b, \end{aligned} \tag{7.31}$$

where A is a $m \times n$ real matrix, $c \in R^n$ and $b \in R^m$. Then, we can use non homogeneous Farkas Lemma to obtain the optimality conditions for linear programming. The interesting fact is that we establish these conditions using only arguments of linear algebra.

Proposition 7.9 (Optimality Conditions for LP) *Problem (7.31) has an optimal solution $x^* \in R^n$ if and only if there exists $u^* \in R^m$ such that*

$$c^T x^* = b^T u^*, \quad Ax^* \geq b, \quad A^T u^* = c, \quad u^* \geq 0. \tag{7.32}$$

Proof Suppose first that Problem (7.31) has an optimal solution $x^* \in R^n$. By definition, we have $Ax^* \geq b$ and we can assert that the following system has no solution:

$$c^T x < c^T x^*, \quad Ax \geq b. \tag{7.33}$$

Letting $\beta = c^T x^*$, by Theorem 7.4 this implies that one of the two following systems must have a solution.

- (II') $b^T u \geq c^T x^*, A^T u = c, u \geq 0,$
- (II'') $b^T u > 0, A^T u = 0, u \geq 0.$

However, system (II'') cannot have solution, because, otherwise, from Gale Theorem we would obtain that the system $Ax \geq b$ has no solution, which would contradict the existence of x^* . Then there must exist $u^* \in R^m$ such that system (II') is satisfied, that is:

$$b^T u^* \geq c^T x^*, \quad A^T u^* = c, \quad u^* \geq 0.$$

Now we can observe that, from $b \leq Ax^*$ and the preceding formula, we obtain

$$b^T u^* \leq u^T Ax^* = (A^T u^*)^T x^* = c^T x^*,$$

so that we can also write $b^T u^* \leq c^T x^*$ and hence, by (II') we have $b^T u^* = c^T x^*$ and (7.32) is satisfied.

Conversely, if (7.32) is satisfied, it follows from Theorem 7.4 that system (7.33) has no solution and hence that the point x^* is an optimal solution. \square

7.7 Notes and References

The basic reference texts for this chapter have been the books [179] and [16]. For the derivation of the theorems of the alternative we have also followed the book [158]. The interested reader can refer to the books [12], [109] and [231] for additional material on optimality conditions and constraint qualifications.

Chapter 8

Basic Concepts on Optimization Algorithms



In this chapter we consider the main structure of the optimization algorithms for the solution of continuous optimization problems on R^n . Then, we discuss the different meanings of convergence and we state some basic definitions of convergence speed.

8.1 Structure of Optimization Algorithms

We refer to the minimization problem

$$\min f(x), \quad x \in S \quad (8.1)$$

where $S \subseteq R^n$ and $f : S \rightarrow R$.

As we know, only in a very few cases we can compute analytically an optimal solution (or establish that the problem has no solution). In most of cases, we must construct an *optimization algorithm*, that is a sequence of well defined steps that, under appropriate assumptions, allows us to determine or to approximate with a good accuracy a “solution” of the problem. In particular, we will restrict our attention to methods that find, in general, *critical points* of the problem, that is points satisfying some necessary optimality condition.

In most of cases, we can exclude that the points determined by the algorithms are local maximizers, but we cannot guarantee that a global minimizer is reached. We cannot even guarantee, in the general case, that a local minimizer can be determined. Moreover, as already discussed, in some difficult problems the algorithm can fail because of the fact that the feasible set is empty or the objective function is unbounded below on S .

When f and S satisfy some (generalized) convexity assumption, we already know that a critical point is also a global solution and thus, if the algorithm determines a critical point, this would solve completely our problem.

In the non convex case, a desirable feature in practical applications, is that we can find a critical point where the objective function is improved with respect to the values obtainable without the use of optimization methods.

In any case, the search for critical points can be inserted within a global optimization scheme, which, for instance, can make use of deterministic or probabilistic criteria for repeating the search from different starting points. But the study of these techniques is out of the scope of this book.

We refer to the conceptual model reported below, where we indicate by Ω the *desired set*, which can be, in general, the set of *critical points* of our problem.

Choose a *starting point* $x_0 \in R^n$ and set $k = 0$.

While $x_k \notin \Omega$

compute $s_k \in R^n$,

compute $x_{k+1} = x_k + s_k$,

set $k = k + 1$.

End While

In connection to this scheme we can note the following points.

- The *starting point* x_0 should be chosen, whenever possible, as the “best” available estimate of an optimal solution. In the constrained case, some techniques assume that a feasible initial point is employed and this may require, as already discussed in Chap. 2, the solution of a feasibility problem through, possibly, another algorithm for the computation of x_0 . However, in the general case, it could be difficult to choose an appropriate starting point and hence we are mostly interested in algorithms that find critical points starting from an arbitrary choice of x_0 in some *known* set X_0 .
- The computation of s_k is obviously the essence of the algorithm. In this phase a local model of the problem is constructed at x_k and s_k is computed through some local algorithm based on this model. In order that the overall algorithm could be considered as *implementable* an important requirement is that s_k can be determined through a *finite* computational procedure. It is also necessary that the local step satisfies suitable conditions that guarantee the termination of the whole process. This typically requires the adoption of some *merit function* that guides the computation of s_k .
- The *termination criterion* consists in checking whether a critical point has been approximated with a given accuracy. From a theoretical point of view, this can be guaranteed if we can show that a point of Ω is reached asymptotically for $k \rightarrow \infty$.

Without loss of generality, we can assume that the algorithm generates an infinite sequence $\{x_k\}$; in fact we could eliminate the stopping criterion by assuming that

$$x_{k+1} = x_k, \quad \text{if } x_k \in \Omega.$$

Thus, the study of the convergence properties of an optimization algorithm consists essentially in the analysis of the behaviour of the infinite sequence $\{x_k\}$, in relation to the objective function and to the feasible set.

In particular, this study may concern the following points:

- existence and uniqueness of limit points of $\{x_k\}$;
- meaning of convergence towards points of Ω ;
- rate of convergence of $\{x_k\}$;
- (sufficient) convergence conditions, in relation to the problem functions;
- overall *computational complexity*.

General indications on these points will be given in next sections; convergence conditions and some issues of computational complexity will be discussed in more detail in connection with the specific classes of algorithms we will describe in the sequel.

8.2 Existence and Uniqueness of Limit Points

In the general case it is quite difficult to guarantee that the sequence generated by an optimization algorithm will converge to a unique point; a basic requirement, however, is that there exist limit points of $\{x_k\}$. As we know, a sufficient condition is that the sequence (or an infinite subsequence of it) remains in a bounded set. As we will see, this condition can be related to the properties of the level sets of the objective function or, more generally, to the level set of some merit function introduced for monitoring the behaviour of the algorithm and for enforcing convergence.

As an example, given $x_0 \in R^n$, let $V : R^n \rightarrow R$ be a continuous function, with (non empty) level set

$$\mathcal{L}_V^0 = \{x \in R^n : V(x) \leq V(x_0)\},$$

such that $\Omega \subseteq \mathcal{L}_V^0$. Then a simple sufficient condition, for the existence of limit points of $\{x_k\}$, is that $x_k \in \mathcal{L}_V^0$ for all k and that \mathcal{L}_V^0 is bounded (and hence compact).

Another example can be that when S is compact and we guarantee that $x_k \in S$, that is when we have constructed a *feasible algorithm* that remains in a compact set.

More complex situations will be considered in the sequel.

Convergence to a unique limit can be imposed in many cases when we can establish the limit

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0, \quad (8.2)$$

which can often be ensured constructively, as we will see, by imposing conditions on the computation of x_{k+1} . We note that condition (8.2) does not guarantee, in general the convergence of the sequence, however we can establish sufficient convergence conditions based on suitable additional assumptions. In particular, we state the following result, whose proof can be found, for instance, in [200], which yields a characterization of limit points.

Proposition 8.1 (Ostrowski) *Let $\{x_k\}$ be a bounded sequence of vectors in R^n and let \mathcal{Z} the set of limit points of $\{x_k\}$. Suppose that*

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

Then, \mathcal{Z} is a compact connected set. □

Condition (8.2) can be weakened to some extent, as shown in the next result (see [87, 186]).

Proposition 8.2 *Let $\{x_k\}$ be a sequence of vectors in R^n such that:*

- (i) *there exists at least an isolated limit point \bar{x} ;*
- (ii) *for every subsequence $\{x_k\}_K$ converging to \bar{x} we have*

$$\lim_{k \in K, k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

Then the sequence $\{x_k\}$ converges to \bar{x} . □

8.3 Convergence Towards Critical Points

We can consider different characterizations of convergence towards desired points of Ω . In particular, from a theoretical point of view, we can distinguish the following cases.

- (a) *There exists an iteration index v such that $x_v \in \Omega$.*

In this case we speak of *finite convergence*. This property can be established only in very special cases. In particular we can define algorithms with finite convergence for the minimization of convex quadratic functions and for the solution of linear programming problems.

- (b) *The sequence $\{x_k\}$ converges to some $\bar{x} \in \Omega$, that is*

$$\lim_{k \rightarrow \infty} x_k = \bar{x}.$$

This requirement cannot be easily enforced in the general case since, typically, we may have different limit points of $\{x_k\}$.

- (c) *Every limit point of $\{x_k\}$ is a critical point.*

We may have different limit points, but each of these points is a critical point. This is the most common convergence property that can be typically established in many algorithms.

- (d) *There exists a limit point $\bar{x} \in \Omega$.*

In this case we may have limit points that are not points of Ω . Although this condition is much weaker than the other convergence properties considered above, we can note that if this property is valid, there must exist a sufficiently large k such that a termination criterion based on an arbitrary tolerance can be satisfied in a finite number of iterations.

8.4 Local and Global Convergence

From the point of view of convergence properties, an important aspect is the dependence of the algorithm on the choice of the starting point and we can distinguish:

- *local convergence properties*, that is properties that hold in some neighborhood of a solution in Ω ;
- *global convergence properties*, that is properties that are satisfied for every choice of the starting point in some prefixed region, such as R^n or \mathcal{L}_S .

We remark that the distinction between these two types of convergence lays essentially in the fact that we speak of local convergence when the neighborhood where convergence can be established is not known a priori (at least in the general case) and we can only establish its existence. It is also to be noted that the terms *global convergence* are not related to the search of global minimizers, but rather express a uniformity property in relation to the choice of starting point in a given (known) region.

In general, we would like to retain as much as possible the choice made through a local model when we estimate that we are proximal to a desired solution, since this may guarantee a fast ultimate convergence speed. However, in order that a neighborhood of the desired solutions can be reached, we must introduce some *globalization technique* that modifies the local choices in order to guarantee convergence towards desired points.

8.5 Convergence Rate

In order to evaluate the *convergence rate* of the sequence x_k generated by an algorithm, we suppose that $\{x_k\}$ converges to some $x^* \in R^n$. Then we define the error e_k in some given norm $\|\cdot\|$:

$$e_k = \|x_k - x^*\|.$$

We consider two possible criteria for evaluating the convergence speed:

- the *Q-convergence*, where we study for $k \rightarrow \infty$, the quotient e_{k+1}/e_k ;
- the *R-convergence*, where we consider the *roots* of the error, or, equivalently, we compare the behaviour of the error with a given law that goes to zero for $k \rightarrow \infty$.

We will confine ourselves to indicate the terminology most frequently used and we refer to the literature, and, in particular, to [200], for a more comprehensive study.

First we introduce the following definitions related to *Q*-convergence.

Definition 8.1 (Convergence Rate: *Q*-Convergence) Let $\{x_k\}$, with $x_k \in R^n$, be a sequence converging to a point $x^* \in R^n$. Then:

- (a) we say that $\{x_k\}$ converges (at least) *Q*-linearly to x^* if there exists $\sigma \in [0, 1)$ such that, for sufficiently large k we have:

$$\|x_{k+1} - x^*\| \leq \sigma \|x_k - x^*\|;$$

- (b) we say that $\{x_k\}$ converges (at least) *Q*-quadratically to x^* if there exists $C > 0$ such that, for sufficiently large k , we have:

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^2;$$

- (c) we say that $\{x_k\}$ converges *Q*-superlinearly to x^* if there holds the limit

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0;$$

(continued)

Definition 8.1 (continued)

- (d) we say that $\{x_k\}$ converges Q -superlinearly to x^* with Q -convergence rate (at least) $p > 1$ if there exists $C > 0$ such that:

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^p.$$

Note that in the preceding definitions we characterize only a *majorization* of the error and hence if (a) holds we cannot exclude that also (b) is valid. Typically, in order to establish some of the preceding conditions we can try to determine a majorization $q(k)$ of the ratio e_{k+1}/e_k , that is

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq q(k)$$

and then characterize the behaviour of $q(k)$ for sufficiently large values of k .

In some cases, however, we are not able to derive this estimate, but we can obtain a majorization $r(k)$ of the error $e(k)$, that is

$$\|x_k - x^*\| \leq r(k),$$

so that the rate of convergence analysis can be carried out by considering the behaviour of $r(k)$ (possibly, by characterizing the Q -convergence of $r(k)$). In this case, we refer to the R -convergence, by assuming an exponential law for $r(k)$.

Definition 8.2 (Convergence Rate: R -Convergence) Let $\{x_k\}$, with $x_k \in R^n$, be a sequence converging to a point $x^* \in R^n$. Then:

- (a) we say that $\{x_k\}$ converges (at least) R -linearly to x^* if there exists $c \in [0, 1)$ such that, for all sufficiently large k , we have:

$$\|x_k - x^*\|^{1/k} \leq c;$$

- (b) we say that $\{x_k\}$ converges R -superlinearly to x^* if there holds the limit

$$\lim_{k \rightarrow \infty} \|x_k - x^*\|^{1/k} = 0;$$

- (c) we say that $\{x_k\}$ converges R -superlinearly to x^* with R -convergence rate (at least) $p > 1$ if there exists $c \in [0, 1)$ such that, for all sufficiently large k , we have

$$\|x_k - x^*\|^{1/p^k} \leq c.$$

The convergence rate depends also on the choice of the starting point and hence, in the definition of the convergence rate, usually we must refer to the worst case in the choice of x_0 in some neighborhood of x^* .

8.6 Basic Concepts of Complexity Analysis

The computational complexity theory, based on the Turing machine to model algorithms, requires to code an optimization problem as “input” to an algorithm and to define the related “size”. The complexity of an algorithm is generally expressed in terms of time and memory requirements, that vary with the size of the coded problem. Most optimization problems with a combinatorial or discrete nature or, for instance, linear programming over the rationals, can be naturally represented as a bit of strings in memory and this easily leads to the notion of “size” of the optimization problem. In the case of continuous optimization problems, the notion of “size” is not immediate, and this motivates the adoption of the “oracle complexity” approach here considered. Note that the oracle-based approach permits even to take into account functions properties (like smoothness and convexity), that affect the practical performance of an algorithm and can not be easily coded as required by the computational complexity theory.

Let us consider a generic class \mathcal{C} of optimization problems

$$\begin{aligned} & \min f(x) \\ & x \in S \subseteq \mathbb{R}^n. \end{aligned} \tag{8.3}$$

We associate to \mathcal{C} an *oracle* \mathcal{O} , which provides the available information $\mathcal{O}(x)$, at a given point x , for a problem in \mathcal{C} . We also have a stopping rule τ_ϵ depending on the degree of precision $\epsilon > 0$. Thus, the complexity study concerns the scheme $(\mathcal{C}, \mathcal{O}, \tau_\epsilon)$.

Just as an example, we could have that

- \mathcal{C} is the class of unconstrained minimization problems of continuously differentiable nonconvex functions;
- $\mathcal{O}(x) = \{f(x), \nabla f(x)\}$, that is, the oracle provides first order information;
- τ_ϵ is defined by $\|\nabla f(x)\| \leq \epsilon$.

An algorithm \mathcal{A} for a scheme $(\mathcal{C}, \mathcal{O}, \tau_\epsilon)$, starting from the given initial point x^0 , generates a sequence of points $\{x^k\}$. In particular, at any iteration k , the new point x^{k+1} is determined using the information provided by the oracle at x^0, x^1, \dots, x^k . The algorithm terminates whenever x^k satisfies τ_ϵ .

The *iteration performance* of an algorithm \mathcal{A} for a scheme $(\mathcal{C}, \mathcal{O}, \tau_\epsilon)$ is a bound on the number of iterations needed to solve any problem belonging to the scheme. Then, this bound is the number of iterations required by Algorithm \mathcal{A} for solving the “worst problem” in $(\mathcal{C}, \mathcal{O}, \tau_\epsilon)$. The bound on the number of iterations will depend

on ϵ and on parameters related to each specific problem of the class \mathcal{C} , that is, number of variables, condition number, Lipschitz constant, etc.

Another measure of performance is the *numerical performance*, which is the number of arithmetical operations required in the worst case. In general, the numerical performance is proportional to the iteration performance and we will consider this latter in the sequel.

The *complexity* of a scheme $(\mathcal{C}, \mathcal{O}, \tau_\epsilon)$ is the performance of the best possible algorithm for the scheme. The main aim of the complexity analysis is to find the best algorithm for different schemes. For a given algorithm, usually an *upper complexity estimate* is determined. Then, the question becomes that of studying whether this latter bound is tight.

8.7 Notes and References

Global convergence issues are deeply investigated in [16] and [196]. The study on the rates of convergence is analyzed in-depth in [200]. Numerical issues of optimization algorithms are considered in classical books like [67], [93], [114]. The complexity of general iterative schemes can be studied in more detail in [193] and [42].

Chapter 9

Unconstrained Optimization Algorithms



In this chapter we first consider the basic properties of unconstrained algorithms and we give a short classification of these techniques. Then we give sufficient conditions of global convergence for the class of line search-based methods.

9.1 Preliminary Concepts

We refer to the unconstrained minimization problem

$$\min f(x), \quad x \in R^n \quad (9.1)$$

where $f : R^n \rightarrow R$ and we denote by $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ the level set corresponding to a given initial point $x_0 \in R^n$.

We assume that f is one or two times continuously differentiable on R^n . However, it is not difficult to relax these requirements by imposing that the differentiability assumptions are valid on some open convex set \mathcal{D} containing \mathcal{L}_0 , provided that all points generated by the algorithm considered, including possible tentative points out of \mathcal{L}_0 , remain in \mathcal{D} .

We know that if \mathcal{L}_0 is bounded (and hence compact, by the continuity of f), then Problem (9.1) has global solutions that satisfy the necessary optimality conditions, and therefore are *stationary points* of f belonging to \mathcal{L}_0 . Actually, if x_0 is not a stationary point, we require also that a stationary point x^* determined by the algorithm is such that $f(x^*) < f(x_0)$.

The conceptual model of an unconstrained optimization algorithm is the same shown in Chap. 8 for the general case. The set Ω is now the set of stationary points in \mathcal{L}_0 , that is

$$\Omega = \{x \in R^n : \nabla f(x) = 0\}.$$

The unconstrained algorithms considered in the sequel attempt to find points of \mathcal{Q} . When f is two times continuously differentiable, we can also define methods that determine points satisfying second order necessary conditions. As already said, in most of cases we can exclude that the points determined by the algorithms are local maximizers, but we cannot guarantee, without some convexity assumption on f , that a global minimizer is attained.

The issues discussed in Sect. 8.1 of Chap. 8 are obviously valid in the unconstrained case. We add a few comments on possible termination criteria. A termination criterion should consist, in principle, in checking whether $\nabla f(x_k) = 0$. In practice, if the gradient of f is available, we could impose, for instance, that the algorithm terminates when we have $\|\nabla f(x_k)\| \leq \varepsilon$ for some given tolerance $\varepsilon > 0$. From a computational point of view this criterion is not entirely satisfactory as it does not take into account the machine precision and the scale used for representing f . However, from a theoretical point of view, the important point is that a termination criterion based on ∇f can be adopted if we can establish that $\|\nabla f(x_k)\|$ can be made arbitrarily small for sufficiently large values of k , at least in some infinite subsequence. This justifies the study of the limit behaviour of our sequence.

Different criteria, which will be discussed in the sequel, must be specified when the derivatives are not available.

9.2 Limit Points

As we already know, the existence of limit points of the sequence $\{x_k\}$ can be established if the points x_k remain in a compact set. If we assume that the level set \mathcal{L}_0 is compact, a sufficient condition for the existence of limit points is that the sequence $\{f(x_k)\}$ is *monotone non increasing*. This also implies the convergence of $\{f(x_k)\}$. More precisely, we can establish the following proposition.

Proposition 9.1 (Existence of Limit Points) *Let $\{x_k\}$ be the sequence generated by the algorithm and assume that f is continuous on R^n and that \mathcal{L}_0 is compact. Suppose that $f(x_{k+1}) \leq f(x_k)$ for all k . Then:*

- (a) $x_k \in \mathcal{L}_0$ for all k ;
- (b) the sequence $\{x_k\}$ has limit points;
- (c) every limit point of $\{x_k\}$ belongs to \mathcal{L}_0 ;
- (d) the sequence $\{f(x_k)\}$ converges.

Proof The assumptions made directly imply that $x_k \in \mathcal{L}_0$ for all k . Then (b) and (c) are immediate consequences of the compactness of \mathcal{L}_0 . The continuity of f and the compactness of \mathcal{L}_0 , imply that f has a global minimizer in R^n and hence that

$f(x_k) \geq \min_{x \in R^n} f(x)$. Thus the sequence is bounded below and, by assumption, it is also monotone non increasing. By a known result on sequences of real numbers, this implies that $\{f(x_k)\}$ converges to a limit. \square

We note that the conditions of the above proposition can be weakened to some extent by requiring that only a subsequence of $\{x_k\}$ is monotonically non increasing and that there holds the limit

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (9.2)$$

This extension will be considered in the sequel in the study of nonmonotone methods.

9.3 Convergence Towards Stationary Points

We can state different characterizations of convergence towards stationary points, by specializing the cases considered in Chap. 8, under the assumption that f is continuously differentiable.

- (a) *There exists an iteration index v such that $\nabla f(x_v) = 0$.*
In this case we speak of *finite convergence*.
- (b) *The sequence $\{x_k\}$ converges to some $\bar{x} \in \Omega$, that is*

$$\lim_{k \rightarrow \infty} x_k = \bar{x} \quad \text{with} \quad \nabla f(\bar{x}) = 0.$$

This requirement cannot be easily enforced and hence, typically, we may have different limit points of $\{x_k\}$.

- (c) *There holds the limit:*

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0. \quad (9.3)$$

As ∇f is continuous, it follows from (9.3) that *every limit point of $\{x_k\}$ is a stationary point*. Then we may have different limit points, but each of these points is a stationary point. If every subsequence of $\{x_k\}$ admits a limit point then (9.3) holds if and only if every limit point is a stationary point. This is the case, for instance, if $x_k \in \mathcal{L}_0$ and \mathcal{L}_0 is compact. We note, however, that (9.3) could be valid even if the sequence has no limit point.

- (d) *There holds the limit:*

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (9.4)$$

This property is obviously weaker than (9.3). In fact, if every subsequence of $\{x_k\}$ admits a limit point, then (9.4) holds *if and only if* there exists a limit point of $\{x_k\}$, which is a stationary point of f , but we may have limit points that are not stationary points of f . In the general case, the limit (9.4) could be valid even if there not exist limit points.

Although (9.4) is much weaker than the other convergence properties considered above, we can note that if this property is valid, by definition of \liminf we can say that, for every $\varepsilon > 0$, there must exist a sufficiently large k such that $\|\nabla f(x_k)\| \leq \varepsilon$, so that a termination criterion based on an arbitrary tolerance $\varepsilon > 0$ can be satisfied in a finite number of iterations.

As already anticipated in Sect. 9.1, we can also give conditions that exclude the existence of limit points that are local maximizers of f . The simplest condition is that the sequence $\{f(x_k)\}$ is strictly decreasing for all sufficiently large k .

9.4 Classification of Unconstrained Algorithms

Unconstrained optimization algorithms for the computation of stationary points can be classified from different points of view. A first obvious distinction is that based on information used in the algorithm.

More specifically we can distinguish:

- methods that use knowledge of the objective function f and of first and second order derivatives, that is gradient ∇f and Hessian matrix $\nabla^2 f$;
- methods that require knowledge of f and ∇f
- derivative-free methods that make use only of function evaluations.

We remark that in all cases we assume that f is at least continuously differentiable, even when we do not use explicitly the knowledge of ∇f .

In each class we have different methods for the computation of s_k , which differs in terms of convergence properties, computational cost and specific criteria for the computation of s_k .

Unconstrained minimization algorithms can be further classified with reference to the basic strategy used at each iteration. In particular, we can distinguish:

- methods based on *line searches*;
- *trust region* methods;
- *direct search* methods.

Line search based methods perform iterations of the form: $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \in R^n$ is the *search direction* and $\alpha_k \in R$ is the *step-size* along d_k . The search direction d_k is typically chosen on the basis of some local model of the objective function and it. can be computed using information available at the current point x_k or else that obtained also at the preceding points; in this case we have *multi-step methods*, where d_k depends also on some past iterates.

The step-size α_k is computed through a *line search*, that is a one-dimensional search along d_k , which has the objective of enforcing convergence towards stationary points of f , under appropriate assumptions on f and d_k , while preserving as much as possible, the local properties on which the choice of d_k was based. Line search methods may require repeated evaluations of f and, possibly, of the derivatives of f , at tentative points along the search direction. Some of the basic monotone line search algorithms will be considered in the next chapter; non monotone line search methods will be introduced in Chap. 24.

Trust region methods consist in computing s_k in the iteration $x_{k+1} = x_k + s_k$, by solving a constrained problem, in which we minimize a model (typically quadratic, possibly non convex) of f over a spherical neighborhood of the current point x_k . Global convergence is enforced by controlling the radius of the neighborhood and by imposing restrictions on the algorithm that solves, approximately, the trust region problem. The study of trust region techniques will be carried out in Chap. 14.

Direct search methods are typically employed when first order derivatives are not available and consist in defining suitable sampling criteria for evaluating the objective function in the region of interest. Some of these techniques make use of derivative-free linesearch algorithms. Algorithms of this type will be studied in Chap. 19.

The basic strategies considered above can be modified and combined in more complex structures by realizing, for instance, *decomposition methods*, where the minimization is performed with respect to different blocks of variables, according to sequential or parallel or mixed schemes. Other strategies, such as *incremental methods* or *on-line methods* may require *function decomposition* techniques where at each step we use partial information on the objective function and its derivatives. Decomposition algorithms will be studied in the sequel.

In the next section we will consider sufficient global convergence conditions for algorithms based on line searches.

9.5 Convergence of Line Search Based Methods

In this section we refer to the following basic scheme.

Choose a starting point $x_0 \in R^n$ and set $k = 0$.

While $x_k \notin \Omega$

1. Compute a search direction $d_k \in R^n$,
2. Determine the step-size $\alpha_k \in R$ along d_k , using a line search technique
3. Compute the new point $x_{k+1} = x_k + \alpha_k d_k$,
4. Set $k = k + 1$.

End While

We are interested in giving sufficient convergence conditions expressed in terms of conditions on d_k and conditions on α_k .

A useful concept introduced in [200], for expressing some conditions, without given attention to minor unessential details, is that given below.

Definition 9.1 (Forcing Function) A function $\sigma : [0, \infty) \rightarrow [0, \infty)$ is a *forcing function* if, for every sequence $\{t_k\}$ with $t_k \in [0, \infty)$, we have that $\lim_{k \rightarrow \infty} \sigma(t_k) = 0$ implies $\lim_{k \rightarrow \infty} t_k = 0$.

A forcing function can be defined, for instance, by letting $\sigma(t) = ct^q$, where $c > 0$ and $q > 0$ are given constants. Any nondecreasing function $\sigma : [0, \infty) \rightarrow [0, \infty)$ such that $\sigma(0) = 0$ and $\sigma(t) > 0$ for $t > 0$ is a forcing function. We note also that the function $\sigma(t) = c > 0$ satisfies vacuously the definition of forcing function, as there not exist sequences such that $\sigma(t_k) \rightarrow 0$. It also easily seen that if σ_1, σ_2 are forcing functions, then the function σ defined by $\sigma(t) = \min\{\sigma_1(t), \sigma_2(t)\}$ is a forcing function. Thus, in particular, we can define a forcing function of the form $\sigma(t) = \min\{a, ct^q\}$, where $a, c, q > 0$.

The conditions given in the next proposition will be frequently used in the sequel whenever the search direction d_k can be related at each step to the gradient $\nabla f(x_k)$.

Proposition 9.2 (Convergence Conditions 1) Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact. Assume that $d_k \neq 0$ for $\nabla f(x_k) \neq 0$ and that the following conditions hold:

- (i) $f(x_{k+1}) \leq f(x_k)$ for every k ;
- (ii) if $\nabla f(x_k) \neq 0$ for all k , we have

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0; \quad (9.5)$$

- (iii) there exists a forcing function $\sigma : [0, \infty) \rightarrow [0, \infty)$ such that, for $d_k \neq 0$ we have

$$\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \geq \sigma(\|\nabla f(x_k)\|). \quad (9.6)$$

(continued)

Proposition 9.2 (continued)

Then, either we have $\nabla f(x_v) = 0$, for some index $v \geq 0$ such that $x_v \in \mathcal{L}_0$ or else the algorithm generates an infinite sequence $\{x_k\}$ such that:

- (a) $x_k \in \mathcal{L}_0$ for all k and $\{x_k\}$; has limit points
- (b) every limit point of $\{x_k\}$ belongs to \mathcal{L}_0 ;
- (c) the sequence $\{f(x_k)\}$ converges;
- (d) $\lim_{k \rightarrow \infty} \nabla f(x_k) = 0$
- (e) every limit point \bar{x} of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.

Proof Condition (i) implies that, if the algorithm does not terminate at a stationary point in \mathcal{L}_0 , it produces an infinite non increasing sequence $\{f(x_k)\}$; then assertions (a), (b) and (c) follow from Proposition 9.1. As σ is a forcing function, assumptions (i) and (ii) imply (d).

Now, if \bar{x} is a limit point of $\{x_k\}$, there must exist a subsequence $\{x_k\}_K$ converging to \bar{x} for $k \in K, k \rightarrow \infty$. As ∇f is continuous, we have

$$\lim_{k \in K, k \rightarrow \infty} \nabla f(x_k) = \nabla f(\bar{x}),$$

and hence, recalling (d), we must have $\nabla f(\bar{x}) = 0$, which establishes (e). \square

The conditions stated in the preceding proposition can be related to conditions on the choice of d_k and α_k . In most of the algorithms employing gradient information considered in the sequel we will assume that d_k satisfies the condition $\nabla f(x_k)^T d_k < 0$, whenever $\nabla f(x_k) \neq 0$, so that d_k is a descent direction and hence we know that f decreases for small values of $\alpha_k > 0$.

Under this assumption, it will be shown in the next chapter that we can define line search procedures such that conditions (i) and (ii) are constructively satisfied.

Condition (iii) imposes limitations on the choice of d_k . In particular, if we choose the forcing function $\sigma(t) = ct$, with $c > 0$, and we assume $\nabla f(x_k)^T d_k < 0$, then condition (9.6) can be expressed into the form

$$\nabla f(x_k)^T d_k \leq -c \|d_k\| \|\nabla f(x_k)\|. \quad (9.7)$$

From a geometrical point of view, condition (9.7) imposes, essentially, that (if $\nabla f(x_k) \neq 0$ for all k) the cosine of the angle θ_k between d_k and $-\nabla f(x_k)$ is bounded away from zero. In fact we have the *angle condition*:

$$\cos(\theta_k) = -\frac{\nabla f(x_k)^T d_k}{\|d_k\| \|\nabla f(x_k)\|} \geq c.$$

We note, in particular, that (9.7) is satisfied with $c = 1$ by assuming

$$d_k = -\nabla f(x_k).$$

In the sequel we will also consider conditions of the form

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^q, \quad \|d_k\| \leq c_2, \quad (9.8)$$

where $c_1 > 0$, $c_2 > 0$ and $q > 0$. It can be easily verified that these conditions imply satisfaction of (9.6).

In Proposition 9.2 we have assumed that the level set \mathcal{L}_0 is compact. In some cases this condition is not satisfied or it could be difficult to establish *a priori* that \mathcal{L}_0 is compact. Thus, it could be of interest restating Proposition 9.2 by omitting the compactness assumption on \mathcal{L}_0 and retaining the other assumptions. It can be easily verified that, in this case, as the sequence $\{f(x_k)\}$ is monotone non increasing either we have $\lim_{k \rightarrow \infty} f(x_k) = -\infty$ (which implies that the problem has no solution), or else the sequence is bounded below. In the latter case we have again that (c) and (d) of Proposition 9.2 hold and that also (e) is verified if we assume that there exist limit points of $\{x_k\}$. But the existence of limit points cannot be ensured.

For some algorithms it could be difficult to establish validity of a condition of the form (9.6) for some forcing function. This typically is the case when d_k is computed on the basis of information related to past iterations. In these cases it could be convenient to relax the conditions on the search direction and to impose stronger conditions on the line search. In particular, we can state the following proposition.

Proposition 9.3 (Convergence Conditions 2) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact. Assume that $d_k \neq 0$ for $\nabla f(x_k) \neq 0$. Let*

$$\cos \theta_k = -\frac{\nabla f(x_k)^T d_k}{\|d_k\| \|\nabla f(x_k)\|},$$

and suppose that the following conditions hold

- (i) $f(x_{k+1}) \leq f(x_k)$ for all k ;
- (ii) if $\nabla f(x_k) \neq 0$ for all k , we have

$$\sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 \cos^2 \theta_k < \infty; \quad (9.9)$$

(continued)

Proposition 9.3 (continued)

(iii) if $\nabla f(x_k) \neq 0$ for all k , we have

$$\sum_{k=0}^{\infty} \cos^2 \theta_k = \infty. \quad (9.10)$$

Then, either we have $\nabla f(x_v) = 0$, for some index $v \geq 0$ such that $x_v \in \mathcal{L}_0$ or else the algorithm generate an infinite sequence $\{x_k\}$ such that:

- (a) $x_k \in \mathcal{L}_0$ for all k and $\{x_k\}$ has limit points;
- (b) every limit point of $\{x_k\}$ belongs to \mathcal{L}_0 ;
- (c) the sequence $\{f(x_k)\}$ converges;
- (d) $\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$;
- (e) there exists a limit point \bar{x} of $\{x_k\}$ such that $\nabla f(\bar{x}) = 0$.

Proof Assertions (a) (b) and (c) follow from (i) and Proposition 9.1. Then assume that the algorithm generates an infinite sequence and suppose, by contradiction, that (d) is false and hence there exists some $\eta > 0$ such that $\|\nabla f(x_k)\| \geq \eta$ for all k . This implies $\|\nabla f(x_k)\|/\eta \geq 1$, so that

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \leq \frac{1}{\eta^2} \sum_{k=0}^{\infty} \|\nabla f(x_k)\|^2 \cos^2 \theta_k.$$

Then, from (9.10) we get a contradiction to (9.9). Therefore, (d) holds and (e) follows from the compactness of \mathcal{L}_0 and the continuity of ∇f . \square

9.6 Exercises

9.1 Prove that conditions (9.8), that is

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^q, \quad \|d_k\| \leq c_2,$$

where $c_1 > 0$, $c_2 > 0$ and $q > 0$ imply satisfaction of the angle condition (9.6).

9.2 In an unconstrained algorithm for minimizing the continuously differentiable function $f : R^n \rightarrow R$, suppose we use at each k a search direction $d_k = -H_k \nabla f(x_k)$, where H_k is a symmetric definite positive matrix such that for all k we have

$$M \geq \lambda_M(H_k) \geq \lambda_m(H_k) \geq m > 0,$$

where $\lambda_M(H_k)$ and $\lambda_m(H_k)$ are, respectively, the largest and the smallest eigenvalue of H_k . Prove that the angle condition (9.6) is satisfied.

9.3 Restate Proposition 9.2, with suitable modifications, omitting the assumption that the level set is compact. (*Follow the indications given after formula (9.8)*).

9.4 Consider the one dimensional unconstrained problem

$$\begin{aligned} \min f(x) &= \frac{1}{2}x^2 \\ x &\in R \end{aligned}$$

whose unique solution is $x^* = 0$. Let x_0 be a starting point such that $|x_0| > 1$. Set $\alpha_k = 2 - \frac{\epsilon_k}{|x_k|}$, with $0 < \epsilon^k < |x_k| - 1$, and

$$x_{k+1} = x_k - \alpha_k f'(x_k) = (1 - \alpha_k)x_k.$$

Show that the generated sequence $\{x_k\}$ is such that

$$f(x_{k+1}) < f(x_k)$$

and

$$\lim_{k \rightarrow \infty} f(x_k) \neq 0.$$

9.7 Notes and References

An introduction to methods based on line searches can be found in classical books like [16], [196], and [12]. As regards the introduction to trust region methods, [50] is an exhaustive reference. Concerning an introduction to direct search methods, we suggest the books [51] and [7].

Chapter 10

Line Search Methods



In this chapter we describe some of the best known line search algorithms employed in the unconstrained minimization of smooth functions. We will restrict our attention to *monotone line search algorithms*. *Non monotone* extensions will be considered in Chap. 24. Here, after a short introduction, we first analyze methods employing derivatives of the objective function and then we describe some derivative-free extensions.

10.1 Basic Features and Classification of Line Searches

As we already know, the main role of a line search algorithm is that of enforcing constructively the convergence of an unconstrained minimization method, when we start from an arbitrary initial point $x_0 \in R^n$.

Given $x_k \in R^n$, we assume that a search direction $d_k \in R^n$ is available and the line search consists in the computation of a *step-size* $\alpha_k \in R$ along d_k . Unless otherwise stated, we assume that the sequence $\{x_k\}$ produced by the algorithm under study is defined by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots \quad (10.1)$$

However, as it will be shown in the sequel, the point $x_k + \alpha_k d_k$ can also be used in some cases only as a reference point in the determination of x_{k+1} .

In *monotone methods*, the computation of α_k , at a given k , is performed through the approximate minimization of the one-dimensional function ϕ defined by

$$\phi(\alpha) = f(x_k + \alpha d_k),$$

which determines the behavior of f along d_k . We have obviously that $\phi(0) = f(x_k)$. As f is assumed to be continuously differentiable, the function ϕ has a continuous first order derivative for all α , which will be denoted by $\dot{\phi}(\alpha)$.

Using the chain rule for composite functions we obtain:

$$\dot{\phi}(\alpha) = \nabla f(x_k + \alpha d_k)^T d_k, \quad (10.2)$$

so that $\dot{\phi}(\alpha)$ is the directional derivative of the function f along d_k , evaluated at the point $x_k + \alpha d_k$. In particular, we have $\dot{\phi}(0) = \nabla f(x_k)^T d_k$.

As already discussed, line search algorithms typically guarantee that the (normalized) directional derivative is eventually driven to zero, that is:

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0. \quad (10.3)$$

Moreover, the algorithm should also ensure at least that the point $x_k + \alpha_k d_k$ remains in the level set \mathcal{L}_0 , corresponding to $f(x_0)$, that is

$$f(x_k + \alpha_k d_k) \leq f(x_0).$$

Additional conditions can be imposed on the line search, in connection with specific unconstrained methods. In particular, it can be required that the derivative $\dot{\phi}(x_k)$ satisfies suitable limitations and that (10.3) is replaced by stronger conditions. In other important cases (nonlinear conjugate gradient methods, non monotone methods, decomposition methods, derivative-free methods) the line search must be often defined in a way that we have also $\lim_{k \rightarrow \infty} \|\alpha_k d_k\| = 0$, which implies, when (10.1) holds, the limit

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

The techniques for enforcing this condition and the consequences on the convergence properties of the minimization algorithms will be described and analyzed in the sequel.

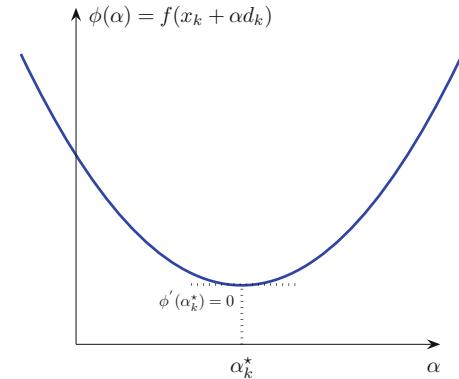
The first criterion proposed in the literature for the computation of α_k consists in requiring that α_k is determined by minimizing f along d_k , that is

$$f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k), \quad \text{for all } \alpha \in R. \quad (10.4)$$

When $x_k \in \mathcal{L}_0$ for all k and \mathcal{L}_0 is compact, then there exists a step-size that solves (10.4), which is often referred to as an *optimal step-size*.

As f is differentiable we must have

$$\dot{\phi}(\alpha_k) = \nabla f(x_k + \alpha_k d_k)^T d_k = 0. \quad (10.5)$$

Fig. 10.1 Optimal step-size

From a geometric point of view, this implies that the gradient evaluated at $x_k + \alpha_k d_k$ must be orthogonal to the search direction d_k .

A line search that computes a step-size satisfying (10.4) is called *exact*, but often this term is used also when only condition (10.5) holds. As shown in Fig. 10.1, in the convex case satisfaction of (10.5) implies also that (10.4) is valid, but, in general, this may be not true for non convex functions. An analytical expression of the optimal step-size can be given only in very special cases. In particular, this is possible when f is a strictly convex quadratic function. In fact we can establish the following proposition.

Proposition 10.1 (Optimal Step-Size in the Quadratic Case) *Let $f(x) = \frac{1}{2}x^T Qx - c^T x$, with Q $n \times n$ symmetric definite positive matrix. Let $x_k, d_k \in \mathbb{R}^n$ with $d_k \neq 0$. Then, the optimal step-size along d_k is given by:*

$$\alpha_k = -\frac{(Qx_k - c)^T d_k}{d_k^T Q d_k} = -\frac{\nabla f(x_k)^T d_k}{d_k^T Q d_k} \quad (10.6)$$

and we have

$$f(x_k + \alpha_k d_k) = f(x_k) + \frac{1}{2} \alpha_k \nabla f(x_k)^T d_k. \quad (10.7)$$

Proof Using Taylor's Theorem we can write, for every $\alpha \in \mathbb{R}$:

$$\phi(\alpha) = f(x_k + \alpha d_k) = f(x_k) + \alpha \nabla f(x_k)^T d_k + \frac{1}{2} \alpha^2 d_k^T Q d_k. \quad (10.8)$$

Therefore, by imposing $\dot{\phi}(\alpha_k) = 0$, recalling the expression of ∇f , we obtain immediately (10.6). Using (10.6) we can write

$$\begin{aligned} f(x_k + \alpha_k d_k) &= f(x_k) + \alpha_k \nabla f(x_k)^T d_k + \frac{1}{2} \alpha_k^2 d_k^T Q d_k \\ &= f(x_k) + \alpha_k \left(\nabla f(x_k)^T d_k - \frac{1}{2} \nabla f(x_k)^T d_k \right), \end{aligned}$$

which establishes (10.7). \square

In the general non quadratic case, even when f is convex, an exact line search could only be performed, in principle, through an iterative method and hence an algorithm employing this line search would not be implementable exactly.

On the other hand, there is no special advantage, neither from a theoretical nor from a computational point of view, in determining at each step the optimal step-size or in computing a good approximation of it. On the contrary, in some cases choosing the optimal step-size could destroy the properties of the search direction in terms of convergence rate. Then, it is much more convenient, in general, to define *inexact* implementable techniques that can guarantee the required convergence properties with an acceptable computational cost. Inexact line search methods consist essentially in defining a set of acceptable α -values, such that:

- a *sufficient decrease* of f is ensured;
- a *sufficiently large* step is effected from x_k .

The acceptance rules that meet these requirement should be satisfied through a finite (possibly small) number of function and gradient evaluations.

Line search techniques can be further classified from different points of view. A first distinction is related to the information required. We can distinguish between

- methods that use information on f and on the derivatives (usually ∇f);
- methods without derivatives, which make use only of function values.

Under appropriate assumptions, it is also possible to define line search algorithms that do not require function evaluations or use only partial information on the objective functions. Some of these techniques, which employ *constant step-sizes* or *step-sizes convergent to zero*, will be discussed later in connection with online or incremental methods.

Another distinction is related to the search strategy; in particular we can distinguish:

- *monotone methods*, which satisfy the condition $f(x_k + \alpha_k d_k) \leq f(x_k)$;
- *non monotone methods*, where the reduction of f is referred to a suitable reference value W_k , in a way that $f(x_k + \alpha_k d_k) \leq W_k \leq f(x_0)$.

When ∇f is available and d_k satisfies the condition

$$\nabla f(x_k)^T d_k < 0, \quad (10.9)$$

we know that d_k is a descent direction and hence there exists an interval $(0, \tilde{\alpha})$ of values $\alpha > 0$ such that $f(x_k + \alpha d_k) < f(x_k)$.

When (10.9) can be established and an inexact line search is performed, we can consider only positive values of α . In the non convex case this obviously does not guarantee that the optimal step-size can be reached. However, under reasonable assumptions on f , a positive value of α satisfying (10.5) exists.

In this chapter we consider first the case where ∇f is available, (10.9) is satisfied and the search is carried out, for each k , starting from an *initial step-size* $\Delta_k > 0$. Under these assumptions, we can distinguish two different types of search techniques:

- *backtracking* methods, where the tentative step-sizes, starting from a sufficiently large initial estimate $\Delta_k > 0$, can only be reduced, until some acceptance condition is satisfied;
- methods that can perform both reductions and *increases* of the tentative step-sizes.

In both cases, an important point is the choice of the initial step-size $\Delta_k > 0$. In many cases an ideal value of Δ_k is defined through the local model used for determining the search direction d_k and hence we can assume, without loss of generality, $\Delta_k = 1$. In these cases the line search should guarantee, at least in a neighborhood of the solution, that the unit step-size can be accepted. In other cases, the choice of d_k does not suggest any specific value for Δ_k and hence it could be convenient to define at each k the initial step-size, on the basis of the past iterations and this may require also increases of the step-size during the search.

In the next sections we will describe and analyze some of the best known monotone line search techniques. Additional results will be given in the sequel, in connection with some specific minimization algorithms.

10.2 Armijo's Method

Armijo's method is one of the first inexact line search algorithms proposed in the literature for the case where, at every given $x_k \in R^n$, with $\nabla f(x_k) \neq 0$, we can compute a descent direction $d_k \in R^n$ such that $\nabla f(x_k)^T d_k < 0$.

It consists, essentially, in a backtracking procedure such that the tentative steps are suitably reduced, if needed, until a sufficient decrease of f is obtained. More specifically, we impose the condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k, \quad (10.10)$$

where $\gamma \in (0, 1)$. Recalling the definition of ϕ , the above inequality can be rewritten into the form:

$$\phi(\alpha_k) \leq \phi(0) + \gamma \alpha_k \dot{\phi}(0).$$

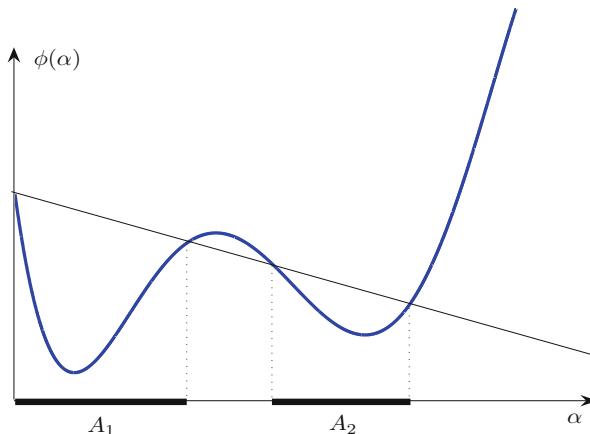


Fig. 10.2 Sufficient decrease criterion in Armijo's method

From a geometrical point of view, this condition imposes that the value $\phi(\alpha_k)$ is not above the line defined by $y = \phi(0) + \gamma\dot{\phi}(0)\alpha$, with slope $\gamma\dot{\phi}(0)$. A sufficiently large step is obtained by imposing a suitable lower bound on the initial step-size and by reducing the tentative step-sizes by a factor $\delta \in (0, 1)$, which is kept constant with respect to k (or remains in a constant interval).

The basic model of Armijo's method is given in the following algorithm, where the index j is a counter of the inner iterations, used only in the convergence proof. The conditions to be imposed on Δ_k will be specified later.

Armijo's Method

```

Data:  $\Delta_k > 0$ ,  $\gamma \in (0, 1)$ ,  $\delta \in (0, 1)$ .
Set  $\alpha = \Delta_k$  and  $j = 0$ .
While  $f(x_k + \alpha d_k) > f(x_k) + \gamma\alpha\nabla f(x_k)^T d_k$ 
    set  $\alpha = \delta\alpha$  and  $j = j + 1$ .
End While
Set  $\alpha_k = \alpha$  and terminate.
```

The sufficient decrease criterion is illustrated in Fig. 10.2, where $A_1 \cup A_2$ is the set of α -values satisfying (10.10).

One iteration of the algorithm is illustrated in Fig. 10.3. In the example considered the initial step-size Δ_k does not satisfy the sufficient decrease criterion; then α is reduced to the value $\delta\Delta_k$, which is accepted, since letting $\alpha_k = \delta\Delta_k$ we have

$$\phi(\alpha_k) < \phi(0) + \gamma\dot{\phi}(0)\alpha_k.$$

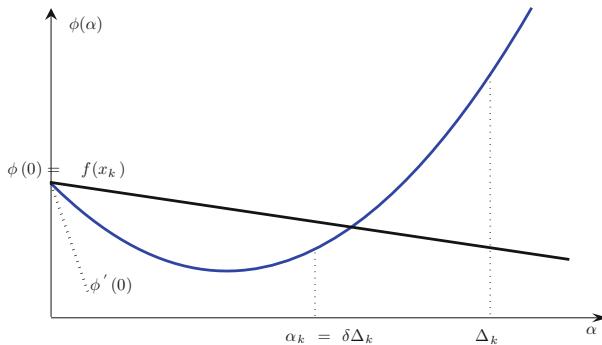


Fig. 10.3 Armijo's method

In the next proposition we prove that Armijo's method terminates in a finite number of j -steps and determines a step-size α_k that satisfies suitable conditions.

Proposition 10.2 (Termination of Armijo's Method) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and let $x_k, d_k \in R^n$ be such that $\nabla f(x_k)^T d_k < 0$. Then Armijo's method terminates in a finite number of inner iterations and determines a step-size $\alpha_k > 0$ such that:*

- (a) $f(x_k + \alpha d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k$;
- (b) one of the following two conditions holds:

$$\begin{aligned} (\text{b}_1) \quad & \alpha_k = \Delta_k; \\ (\text{b}_2) \quad & \alpha_k \leq \delta \Delta_k \quad \text{and} \quad f(x_k + \frac{\alpha_k}{\delta} d_k) > f(x_k) + \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k. \end{aligned}$$

Proof First we show that the *while cycle* terminates. Reasoning by contradiction, we can assume that the acceptance condition is not satisfied for a finite value of j . Therefore, at every j , we have:

$$\frac{f(x_k + \Delta_k \delta^j d_k) - f(x_k)}{\Delta_k \delta^j} > \gamma \nabla f(x_k)^T d_k.$$

As $\delta_j < 1$, we have $\lim_{j \rightarrow \infty} \delta^j = 0$. Taking limits for $j \rightarrow \infty$, we obtain:

$$\nabla f(x_k)^T d_k \geq \gamma \nabla f(x_k)^T d_k,$$

which contradicts the assumptions $\nabla f(x_k)^T d_k < 0$ and $\gamma < 1$, so that assertion (a) must hold.

Now, the instructions of the algorithm imply that (b₁) holds when the initial step-size is accepted and hence $\alpha_k = \Delta_k$. If Δ_k is not accepted we have necessarily that $\alpha_k \leq \delta\Delta_k < \Delta_k$. In this case we have

$$f(x_k + \frac{\alpha_k}{\delta}d_k) - f(x_k) > \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k,$$

for, otherwise, the value α_k/δ should have been accepted. Thus, one of the two conditions (b₁) and (b₂) must be true. \square

On the basis of the preceding proposition, the step-size computed by Armijo's method can be expressed in the form:

$$\max \left\{ \alpha \in R : \alpha = \Delta_k \delta^j, \phi(\alpha) \leq \phi(0) + \gamma \dot{\phi}(0)\alpha, \quad j = 0, 1, \dots \right\}.$$

The convergence is established in the next proposition where we specify a *sufficiently large* lower bound on the initial step-size Δ_k .

Proposition 10.3 (Convergence of Armijo's Method) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n . Assume that the level set \mathcal{L}_0 is compact and that condition $\nabla f(x_k)^T d_k < 0$ is satisfied for every k . Assume also that the initial step-size $\Delta_k \in R^+$ satisfies the condition:*

$$\Delta_k \geq \frac{1}{\|d_k\|} \sigma \left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right), \quad (10.11)$$

where $\sigma : R^+ \rightarrow R^+$ is a forcing function. Then Armijo's method terminates and determines a step-size $\alpha_k > 0$ such that the sequence defined by

$$x_{k+1} = x_k + \alpha_k d_k$$

satisfies the conditions:

- (c₁) $f(x_{k+1}) < f(x_k)$;
- (c₂) $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0$.

Proof As proved in the preceding proposition, Armijo's method terminates and yields $\alpha_k > 0$ such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k. \quad (10.12)$$

Then assertion (c₁) follows immediately from this condition and the assumption $\nabla f(x_k)^T d_k < 0$. Now we prove that assertion (c₂) holds. Using (10.12), we can write:

$$f(x_k) - f(x_{k+1}) \geq \gamma \alpha_k |\nabla f(x_k)^T d_k| = \gamma \alpha_k \|d_k\| \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \quad (10.13)$$

By the compactness of \mathcal{L}_0 and the continuity of f we know that the monotonically decreasing sequence $\{f(x_k)\}$ has a limit, and hence from (10.13) we obtain:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} = 0. \quad (10.14)$$

Reasoning by contradiction, assume now that assertion (c₂) is false. As the sequence $\{\nabla f(x_k)^T d_k / \|d_k\|\}$ is bounded there must exist a subsequence (that we redefine as $\{x_k\}$) and a number $\eta > 0$, such that:

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = -\eta < 0. \quad (10.15)$$

Then, by (10.14) we have:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0. \quad (10.16)$$

As $x_k \in \mathcal{L}_0$ and the sequence $\{d_k / \|d_k\|\}$ is bounded there must exist subsequences (which we redefine again $\{x_k\}$ and $\{d_k\}$), such that

$$\lim_{k \rightarrow \infty} x_k = \hat{x}, \quad \lim_{k \rightarrow \infty} \frac{d_k}{\|d_k\|} = \hat{d}. \quad (10.17)$$

Then, from (10.15) and (10.17), as ∇f is continuous, we get:

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\hat{x})^T \hat{d} = -\eta < 0. \quad (10.18)$$

Now suppose that (b₁) of Proposition 10.2 holds for some infinite subsequence. Then by (10.11) we have :

$$\Delta_k \|d_k\| \geq \sigma \left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right),$$

so that, taking limits on the subsequence considered we have that, by (10.16), the corresponding subsequence of numbers $|\nabla f(x_k)^T d_k|/\|d_k\|$ converges to zero and this contradicts (10.18).

Therefore, we can assume that for sufficiently large values of k , say $k \geq \hat{k}$, we must have $\alpha_k < \Delta_k$, so that assertion (b₂) of Proposition 10.2 must hold. It follows that for $k \geq \hat{k}$ we have:

$$f(x_k + \frac{\alpha_k}{\delta} d_k) - f(x_k) > \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k. \quad (10.19)$$

Using the mean value theorem, we can write

$$f(x_k + \frac{\alpha_k}{\delta} d_k) = f(x_k) + \frac{\alpha_k}{\delta} \nabla f(z_k)^T d_k, \quad (10.20)$$

with $z_k = x_k + \theta_k(\alpha_k/\delta)d_k$, where $\theta_k \in (0, 1)$. By substituting (10.20) into (10.19), for $k \geq \hat{k}$, we have: $\nabla f(z_k)^T d_k > \gamma \nabla f(x_k)^T d_k$. Thus, dividing both members of this inequality by $\|d_k\|$, we get

$$\frac{\nabla f(z_k)^T d_k}{\|d_k\|} > \gamma \frac{\nabla f(x_k)^T d_k}{\|d_k\|}. \quad (10.21)$$

On the other hand, by (10.17) and (10.16) we have

$$\lim_{k \rightarrow \infty} z_k = \lim_{k \rightarrow \infty} \left(x_k + \theta_k \frac{\alpha_k}{\delta} d_k \right) = \hat{x}.$$

Then, taking limits for $k \rightarrow \infty$, from (10.21) we get $\nabla f(\hat{x})^T \hat{d} \geq \gamma \nabla f(\hat{x})^T \hat{d}$ and hence, by (10.18), it follows that $\eta \leq \gamma \eta$, which contradicts the assumption $\gamma < 1$.

We can conclude that assumption (10.15) yields a contradiction in all cases, so that (c₂) must be true. \square

We shortly discuss here the choice of the parameters Δ_k, γ, δ considered in the model algorithm of Armijo's method; other indications on computational aspects will be given in the appendix to this chapter.

In principle, the initial step-size Δ_k can be guaranteed to be sufficiently large by choosing arbitrarily a forcing function and then choosing Δ_k that satisfies (10.11). However, in many cases, as already remarked, a unit initial step-size would be desirable. In this case, in order to ensure that (10.11) is satisfied we must show the existence of a function σ such that the condition:

$$\|d_k\| \geq \sigma \left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right) \quad (10.22)$$

is satisfied. Under the assumption that \mathcal{L}_0 is compact, a sufficient condition on d_k , which implies (10.22), is the condition

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^p, \quad c_1, p > 0. \quad (10.23)$$

In fact, as $\nabla f(x_k)^T d_k < 0$, using (two times) Schwarz inequality we can write:

$$\left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right)^p \leq \|\nabla f(x_k)\|^p \leq \frac{1}{c_1} |\nabla f(x_k)^T d_k| \leq \frac{M}{c_1} \|d_k\|, \quad (10.24)$$

where M denotes an upper bound on $\|\nabla f(x)\|$ for $x \in \mathcal{L}_0$. Thus condition (10.22) holds with

$$\sigma(t) = t^p c_1 / M.$$

The parameter $\gamma \in (0, 1)$ determines the slope of the line that defines the condition of sufficient decrease. It is usually assumed that $\gamma < 1/2$, and this guarantees that the optimal step-size is accepted in the quadratic case.

This follows immediately from (10.7) of Proposition 10.1. In fact, if f is quadratic and α_k is the optimal step-size, Eq. (10.7) implies, for $\gamma \leq 1/2$:

$$f(x_k + \alpha_k d_k) = f(x_k) + \frac{1}{2} \alpha_k \nabla f(x_k)^T d_k \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k. \quad (10.25)$$

The condition $\gamma < 1/2$ is also important in some methods in the study of the convergence rate. In practice, the values of γ are usually much smaller than $1/2$ and of the order 10^{-3} – 10^{-4} .

The parameter $\delta \in (0, 1)$ in Armijo's method has been assumed constant, and a possible value could be, for instance, $\delta = 0.5$. However, it can be easily verified that the convergence of the method is preserved even if we make use a variable factor $\delta(k, j)$ in a constant interval, that is, if $\delta(k, j) \in [\delta_l, \delta_u]$, with $0 < \delta_l < \delta_u < 1$. Typical values of these bounds could be, for instance, $\delta_l = 0.1$ and $\delta_u = 0.9$.

The computation of $\delta(k, j)$ can be performed through a safeguarded interpolation technique and this can be useful in many methods for reducing the number of function evaluations during the line search.

The convergence results obtained for Armijo's method can easily be extended to the convergence analysis of other algorithms where we do not assume necessarily that $x_{k+1} = x_k + \alpha_k d_k$ for all k .

A first observation is that the proof of Proposition 10.3 remains substantially unchanged if x_{k+1} is chosen arbitrarily (even not along the ray defined by the search direction d_k), provided that $f(x_{k+1}) \leq f(x_k + \alpha_k^A d_k)$, where α_k^A is the step-size computed through Armijo's method. This implies, in particular that acceptable values for α are those such that $f(x_k + \alpha d_k) \leq f(x_k + \alpha_k^A d_k)$. Thus Proposition 10.3 proves also the convergence of the exact line search, since the optimal step-size obviously satisfies the above inequality.

Remark 10.1 A second important observation is that the convergence results can be extended to the case where the line searches are performed only in correspondence to some subsequence, denoted by $\{x_k\}_K$, of an arbitrary sequence $\{x_k\}$, provided that we can still establish the convergence of $\{f(x_k)\}$. \square

10.3 Step-Length Extension and Goldstein Conditions

We have already observed that in some cases the choice of d_k does not suggest an ideal step-size, so that it could be convenient to choose an initial step-length (on the basis of previous iterations or through a quadratic approximation), which could not satisfy a condition of the form (10.11). In these cases, in order to guarantee the convergence of the line search we must check whether the initial step-size is sufficiently large and possibly we must increase the initial estimate on the basis of some criterion.

Then Armijo's method must be modified by introducing suitable rules for extending the step-length. One of the simplest criteria can be that of imposing the following acceptance conditions.

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k \\ f(x_k + \mu_k \alpha_k d_k) &\geq \min [f(x_k + \alpha_k d_k), f(x_k) + \gamma \mu_k \alpha_k \nabla f(x_k)^T d_k] \end{aligned} \quad (10.26)$$

where $\mu_k \in [\mu_l, \mu_u]$ and $1 < \mu_l \leq \mu_u$. The second condition in (10.26) is essentially a condition implying that the step-length is *sufficiently large*.

As we will suggest in the sequel, this condition can be used in the proof of Proposition 10.3, in place of condition (10.19).

A different criterion, which permits to accept an arbitrary initial estimate, without performing a new function evaluation, can be based on the so called *Goldstein conditions*, actually introduced before than Armijo's proposal.

Assuming $\nabla f(x_k)^T d_k < 0$, and letting $0 < \gamma_1 < \gamma_2 < 1/2$, Goldstein conditions are given by

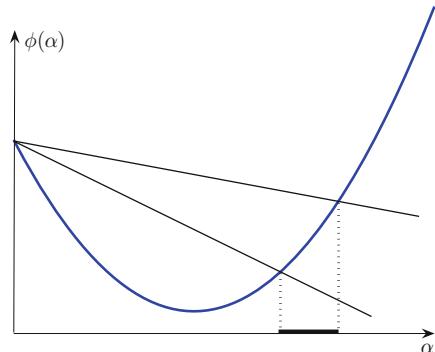
$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma_1 \alpha_k \nabla f(x_k)^T d_k \quad (10.27)$$

$$f(x_k + \alpha_k d_k) \geq f(x_k) + \gamma_2 \alpha_k \nabla f(x_k)^T d_k. \quad (10.28)$$

The geometrical meaning of these conditions is illustrated in Fig. 10.4, where we indicate the interval of acceptable α -values.

We can note that (10.27) is the same condition of sufficient decrease used in Armijo's method, while, as $\gamma_2 < 1$, inequality (10.28) guarantees that the step-size is not too small and has essentially the same role of the second condition in (10.26). Now, however, we do not need necessarily a new function evaluation for accepting (when possible) the initial step-size.

Fig. 10.4 Goldstein conditions



Under the assumption stated, it can be shown that there exists an interval $[\alpha_l, \alpha_u]$ of values that satisfy Goldstein condition and it can be easily verified that a convergence proof can be carried out along the same lines followed in the proof of Proposition 10.3. We must only note that no condition must be imposed on the initial step-size and that (10.19) can be replaced, when required, by condition (10.28).

We could easily define an algorithm that computes α_k in a way that Goldstein conditions are satisfied. However, in some instances, determining a point in the interval of acceptable α -values could be unnecessarily expensive. Therefore, we prefer to combine Goldstein conditions with conditions (10.26), in the attempt of retaining the advantages of both approaches.

We define the following model algorithm.

Armijo-Goldstein Method

Data: $\Delta_k > 0, 0 < \gamma_1 < \gamma_2 < 1/2, 0 < \delta < 1$.

Set $\alpha = \Delta_k$, $j = 0$ and $h = 0$.

While

$$\begin{aligned} f(x_k + \alpha d_k) &> f(x_k) + \gamma_1 \alpha \nabla f(x_k)^T d_k \\ \text{set } \alpha &= \delta \alpha \text{ and } j = j + 1 \end{aligned}$$

End While

If $\alpha < \Delta_k$ set $\alpha_k = \alpha$ and **terminate**.

While

$$f(x_k + \alpha d_k) < f(x_k) + \gamma_2 \alpha \nabla f(x_k)^T d_k$$

$$f(x_k + \frac{\alpha}{\delta} d_k) < \min\{f(x_k + \alpha d_k), f(x_k) + \gamma_1 \frac{\alpha}{\delta} \nabla f(x_k)^T d_k\}$$

set $\alpha = \alpha/\delta$ and $h = h + 1$.

End While

Set $\alpha_k = \alpha$ and **terminate**. □

We note that the first while cycle is essentially equivalent to Armijo's method and hence it terminates for a finite value of the counter j . If $\alpha_k < \Delta_k$ the convergence proof is the same as that given for Armijo's method, since we do not need step expansions and a local minimizer is bounded above by Δ_k .

The expansion step starts only when Δ_k satisfies the sufficient decrease condition and the second while cycle necessarily terminates if f is bounded below. In fact, if $h \rightarrow \infty$ we have $|\alpha|/\delta \rightarrow \infty$ and this would imply $\phi(\alpha) \rightarrow -\infty$. Then we can state the following proposition.

Proposition 10.4 (Convergence of Armijo-Goldstein Algorithm) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n . Assume that the level set \mathcal{L}_0 is compact and that condition $\nabla f(x_k)^T d_k < 0$ is satisfied for every k . Let $\{x_k\}$ be a sequence defined by*

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

where the step-length α_k is computed by means of Armijo-Goldstein method. Then we have

- (c₁) $f(x_{k+1}) < f(x_k)$;
- (c₂) $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0$.

Proof Reasoning as in the proof of Proposition 10.3, we can assert that (c₁) holds. We can assume, by contradiction, that (c₂) is false and hence that there exist subsequences that we relabel $\{x_k\}$ and $\{d_k\}$, such that

$$\lim_{k \rightarrow \infty} x_k = \hat{x}, \quad \lim_{k \rightarrow \infty} \frac{d_k}{\|d_k\|} = \hat{d}$$

and that

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\hat{x})^T \hat{d} = -\mu < 0. \quad (10.29)$$

Now, if for some infinite subsequence the algorithm terminates with $\alpha_k < \Delta_k$ we can repeat exactly the same proof given for Proposition 10.3.

Then we can consider only the case where $\{x_k\}$ is an infinite sequence such that $\alpha = \Delta_k$ at the start of the second while cycle.

The instruction of the algorithm imply that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma_1 \alpha_k \nabla f(x_k)^T d_k, \quad (10.30)$$

and that one of the following condition holds:

$$f(x_k + \alpha_k d_k) \geq f(x_k) + \gamma_2 \alpha_k \nabla f(x_k)^T d_k, \quad (10.31)$$

$$f(x_k + \frac{\alpha_k}{\delta} d_k) \geq f(x_k) + \gamma_1 \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k, \quad (10.32)$$

$$f(x_k + \frac{\alpha_k}{\delta} d_k) \geq f(x_k + \alpha_k d_k). \quad (10.33)$$

Using the mean value theorem¹ in each of the three conditions (10.31), (10.32) and (10.33), taking limits for $k \rightarrow \infty$, and reasoning as in the proof of Proposition 10.3, we get a contradiction to (10.29). This establishes (c₂) and concludes the proof. \square

As already remarked in connection with Armijo's method, we can modify the Armijo-Goldstein algorithm by replacing the fixed factor $\delta \in (0, 1)$ with a variable factor $\delta(j, k)$ such that $\delta(j, k) \in [\delta_l, \delta_u]$, $0 < \delta_l \leq \delta_u < 1$. Similarly, the expansion factor $1/\delta$ can be replaced by some variable factor $\mu(h, k)$ such that $\mu(h, k) \in [\mu_l, \mu_u]$, $1 < \mu_l \leq \mu_u$.

10.4 Wolfe Conditions

In some minimization methods the line search algorithm should guarantee that the derivative of ϕ at α_k , that is $\dot{\phi}(\alpha_k) = \nabla f(x_k + \alpha_k d_k)^T d_k$, satisfies suitable limitations. Acceptability conditions that take this requirement into account are known as *Wolfe conditions*. Assuming $\nabla f(x_k)^T d_k < 0$, we can define the two criteria given below.

¹ We note that the mean value theorem can be used in both members of (10.33), thus obtaining

$$f(x_k) + \frac{\alpha_k}{\delta} \nabla f(x_k + \theta_k \frac{\alpha_k}{\delta} d_k)^T d_k \geq f(x_k) + \alpha_k \nabla f(x_k + \xi_k \alpha_k d_k)^T d_k,$$

where $\theta_k, \xi_k \in (0, 1)$.

Weak Wolfe conditions (W1)

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k \quad (10.34)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma \nabla f(x_k)^T d_k. \quad (10.35)$$

Strong Wolfe conditions (W2)

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k \quad (10.36)$$

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq \sigma |\nabla f(x_k)^T d_k| \quad (10.37)$$

where $\gamma \in (0, 1/2)$ and $\sigma \in (\gamma, 1)$.

In both conditions we impose, through (10.34) or (10.36), a sufficient decrease of f , using the same acceptance rule appearing in Armijo-Goldstein criteria. Conditions (10.35) and (10.37) guarantee that the step-length is not too small by imposing bounds on the slope of ϕ at α_k , which is required to be significantly greater than $\dot{\phi}(0)$.

In particular, (10.35) imposes that the slope of ϕ at α_k either is non negative or, when negative, has a modulus not grater than $|\sigma \nabla f(x_k)^T d_k|$. The geometrical meaning of this condition is illustrated in Fig. 10.5.

Condition (10.37) requires that ϕ is sufficiently flat at α_k , as the absolute value of the slope should be inferior to $\sigma |\nabla f(x_k)^T d_k|$. Clearly conditions (W2) are more restrictive than (W1).

It can be shown that if we have $1 > \sigma > \gamma > 0$ then there exists an interval of acceptable α -values for the strong Wolfe conditions (and hence also for the weak conditions). Condition (W2) is illustrated in Fig. 10.6.

Fig. 10.5 Weak Wolfe conditions

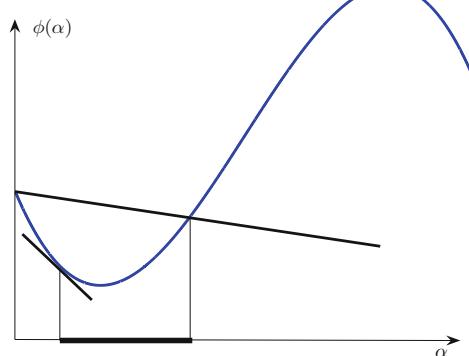
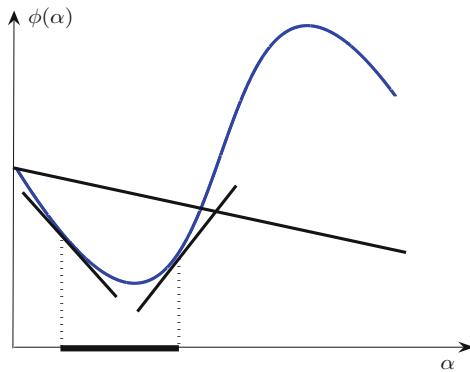


Fig. 10.6 Strong Wolfe conditions



We note that line searches based on Wolfe conditions require that, at each tentative step-size, both the function value and the first order derivatives are evaluated. This makes each step more expensive, in comparison with Armijo-Goldstein criteria, but gradient information can be used to improve the efficiency of the search. In particular, cubic interpolation techniques can be conveniently adopted.

We can establish a convergence result similar to that given in Proposition 10.3. As satisfaction of the strong conditions (W2) implies satisfaction of the weak conditions (W1), we can state a convergence result only with reference to (W1).

Proposition 10.5 (Convergence of Wolfe's Method) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n . Assume that the level set \mathcal{L}_0 is compact and that condition $\nabla f(x_k)^T d_k < 0$ is satisfied for every k . Let $\{x_k\}$ be a sequence defined by $x_{k+1} = x_k + \alpha_k d_k$, where the step-length α_k is computed in a way that the weak Wolfe condition (W1) are satisfied. Then we have*

- (c₁) $f(x_{k+1}) < f(x_k)$;
- (c₂) $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0$.

Proof Reasoning as in the proof of Proposition 10.3 we can easily show that (c₁) holds and we have:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| \left| \frac{\nabla f(x_k)^T d_k}{\|d_k\|} \right| = 0. \quad (10.38)$$

Assume now, by contradiction, that (c₂) is false. Then there must exist a subsequence (which we relabel {x_k}), such that

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = -\eta < 0. \quad (10.39)$$

Therefore by (10.38) we have, in the same subsequence:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0. \quad (10.40)$$

Adding the term $-\nabla f(x_k)^T d_k$ to both members of (10.35), we get

$$\nabla f(x_k + \alpha_k d_k)^T d_k - \nabla f(x_k)^T d_k \geq (\sigma - 1) \nabla f(x_k)^T d_k,$$

whence, recalling that $\nabla f(x_k)^T d_k < 0$ and $\sigma < 1$, we get

$$|\nabla f(x_k)^T d_k| \leq \frac{1}{1 - \sigma} \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\| \|d_k\|, \quad (10.41)$$

which implies

$$\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \leq \frac{1}{1 - \sigma} \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\|. \quad (10.42)$$

As \mathcal{L}_0 is compact the continuous function ∇f is uniformly continuous on \mathcal{L}_0 . Therefore, as $x_k \in \mathcal{L}_0$, the limit (10.40) implies that, taking limits for $k \rightarrow \infty$, we get a contradiction to (10.39); thus (c₂) must be true. \square

10.5 Derivative-Free Line Searches

In this section we consider some implementable, derivative-free, inexact line search techniques employing only function values, which will be used in Chap. 19 for globalizing derivative-free methods. Our objective is that of constructing line search algorithms that retain, essentially, the same properties of the techniques employing derivative information studied in the preceding sections. We suppose that our problem is that of minimizing a continuously differentiable function $f : R^n \rightarrow R$ and that {x_k} is a sequence of points in R^n , such that at each x_k we perform a line search along some direction $d_k \in R^n$.

We require that $f(x_{k+1}) \leq f(x_k)$ or, at least, that $f(x_{k+1}) \leq f(x_0)$ and that, in the limit, we have

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

Moreover, we would also satisfy the condition

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0,$$

which is an important requirement when convergence depends on the searches performed in some finite set of different iterations.

Under these conditions, by an appropriate choice of the search directions, we can construct algorithms with global convergence towards critical points of f , without requiring the evaluation of the derivatives.

We observe preliminarily that when derivative information is not available, in general we cannot establish analytically whether a given direction d_k is a descent direction at x_k . Therefore, we must introduce criteria for choosing the sign of the search direction and for terminating the search when the tentative step-size becomes unacceptably small. In fact, we cannot exclude that $\alpha = 0$ is a local minimizer of the function

$$\phi_k(\alpha) = f(x_k + \alpha d_k).$$

In alternative to bidirectional searches, we can also perform only searches for $\alpha \geq 0$, but along a suitable set of different search directions.

We describe first two basic schemes employing a bidirectional search:

- a backtracking Armijo-type scheme;
- an algorithm of Goldstein-Armijo type with possible step expansion.

Then we show the simple modifications that we can introduce in case of searches limited to nonnegative step-size.

10.5.1 Backtracking Armijo-Type Derivative-Free Algorithms

One of the simplest Armijo-type derivative-free line searches is based on the adoption of an acceptance condition of the form

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \gamma \alpha_k^2 \|d_k\|^2, \quad (10.43)$$

where $\gamma > 0$ and the *sufficient decrease* is imposed through a “parabolic” term. When the sequence of function values converges, this condition also enforces the limit $\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0$.

Starting from a “sufficiently large” initial step-size $\Delta_k > 0$, the algorithm terminates in a finite number of steps, by computing a step-size $\alpha_k \neq 0$ that yields a sufficient reduction of f or by giving in output $\alpha_k = 0$, which implies that $x_{k+1} = x_k$ at the current iteration. In the latter case the search along d_k has failed.

A conceptual algorithm model is given below

Algorithm DFALS: Derivative-Free Armijo-Type Linesearch

Data. $\Delta_k > 0$, $\gamma > 0$, $\delta \in (0, 1)$, $\rho_k > 0$.

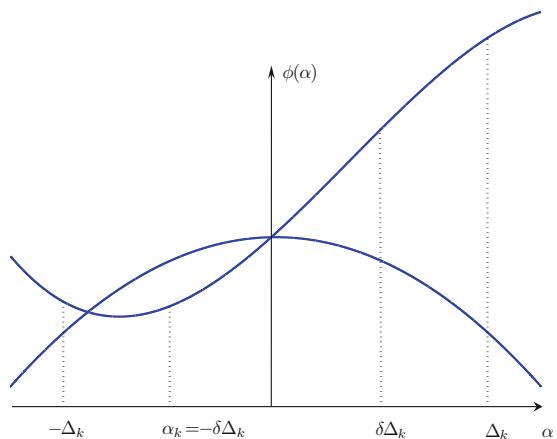
1. Set $\alpha = \Delta_k$.
2. **While** $f(x_k + u\alpha d_k) > f(x_k) - \gamma\alpha^2\|d_k\|^2$ for $u = \pm 1$, **do**
 - If** $\alpha\|d_k\| < \rho_k$ **then**
 - set $\eta_k = \alpha$, $\alpha_k = 0$ and **terminate**.
 - Else**
 - set $\alpha = \delta\alpha$.
 - End If****End while**
3. Set $\alpha_k = u_k\alpha$, where $u_k \in \{-1, 1\}$ is the value such that the condition of sufficient reduction (10.43) is satisfied and **terminate**.

It is easily seen that the algorithm is well defined and terminates in a finite number of steps, since at each inner iteration of the while cycle the tentative step-size is reduced by a factor $\delta < 1$.

The steps performed by the algorithm are illustrated in Fig. 10.7, with reference to a case when a step-size $\alpha_k \neq 0$ is computed in two inner iterations.

The convergence properties are established in the next proposition.

Fig. 10.7 Derivative-free linesearch



Proposition 10.6 (Convergence of Algorithm DFALS) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact.*

For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that:

- (i) *for every k , we have*

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

where $d_k \neq 0$ and the step-size α_k is computed by employing Algorithm DFALS;

- (ii) *the initial step-size is such that $\Delta_0 \geq a/\|d_0\|$, $a > 0$;*
- (iii) *we have $\rho_k \rightarrow 0$ for $k \rightarrow \infty$.*

Then, we have

- (c₁) *$x_k \in \mathcal{L}_0$ for all k ;*
- (c₂) *the sequences $\{f(x_k)\}$ converges to a limit;*
- (c₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c₄) $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0$.

Proof The instructions of the algorithm guarantee that $f(x_{k+1}) \leq f(x_k)$, so that (c₁) holds and hence, taking into account the compactness of \mathcal{L}_0 , and the continuity of f , also (c₂) must hold. Thus the convergence of $\{f(x_k)\}$ and the instructions of the algorithm imply that (c₃) is satisfied.

In order to prove (c₄), let us assume, by contradiction, that (c₄) is false. Then, by (c₁) and the compactness of \mathcal{L}_0 , we can find an infinite subsequence (which we relabel again $\{x_k\}$), such that

$$\lim_{k \rightarrow \infty} x_k = \bar{x}, \quad \lim_{k \rightarrow \infty} d_k / \|d_k\| = \bar{d}, \quad \text{where } \|\bar{d}\| = 1$$

and

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} \neq 0. \quad (10.44)$$

Now, suppose first there exists \bar{k} such that, in the subsequence considered), Algorithm DFALS terminates with $\alpha_k = 0$, for all $k \geq \bar{k}$. In this case, for all $k \geq \bar{k}$ we have

$$f(x_k + \eta_k d_k) > f(x_k) - \gamma_1 \eta_k^2 \|d_k\|^2, \quad f(x_k - \eta_k d_k) > f(x_k) - \gamma_1 \eta_k^2 \|d_k\|^2,$$

where $\eta_k \|d_k\| < \rho_k$.

Using the Mean Value Theorem, we have that there exist points

$$u_k = x_k + \zeta_k \eta_k d_k \quad \text{and} \quad v_k = x_k - \beta_k \eta_k d_k,$$

with $\zeta_k, \beta_k \in (0, 1)$, such that

$$\frac{\nabla f(u_k)^T d_k}{\|d_k\|} > -\gamma_1 \eta_k \|d_k\|,$$

$$\frac{\nabla f(v_k)^T d_k}{\|d_k\|} < \gamma_1 \eta_k \|d_k\|.$$

As $\eta_k \|d_k\| \leq \rho_k$ and $\rho_k \rightarrow 0$ for $k \rightarrow \infty$, we have that $u_k \rightarrow \bar{x}$ and $v_k \rightarrow \bar{x}$, so that we get in the limit

$$\nabla f(\bar{x})^T \bar{d} = 0$$

and this contradicts (10.44).

Therefore we must assume there exists an infinite subsequence (denoted again $\{x_k\}$) such that $\alpha_k \neq 0$ for all k . We can distinguish two different cases.

Case (a) We have $|\alpha_k| = \Delta_k$ for an infinite subsequence, so that $|\alpha_k| \geq a/\|d_k\|$. However, by (c₃) we must have $\alpha_k \|d_k\| \rightarrow 0$ and hence we get a contradiction.

Case (b) We have $0 < |\alpha_k| < \Delta_k$ for sufficiently large k . Then we can repeat a similar reasoning to that followed in the case $\alpha_k = 0$, using the Mean Value Theorem, replacing η_k with α_k/δ_k and taking into account assertion (c₃). Thus we get again a contradiction to (10.44). \square

A possible disadvantage of Algorithms DFALS could be that the condition imposed on the starting tentative step-size could determine, in general, an inefficient choice. In fact, if we impose a condition of the form $\Delta_k \geq a/\|d_k\|$, by (c₃) we have that the first tentative step-size Δ_k will be never accepted for sufficiently large k .

To overcome this limitation, in the next paragraph we will define a derivative-free Armijo-Goldstein scheme, where the initial tentative step-size can be chosen arbitrarily, on the basis of the preceding iterations, but it can be increased, if needed, during the search, in order to guarantee the adoption of a sufficiently large step-size.

10.5.2 Derivative-Free Linesearches with Step Extension

We give below a conceptual version of a derivative-free Armijo-Goldstein-type algorithm with arbitrary initial tentative step-size and Goldstein-type criteria for increasing the step-size.

Derivative-Free Armijo-Goldstein-Type Linesearch (DFAGLS)

Data. $\Delta_k > 0$, $\gamma_2 > \gamma_1 > 0$, $\delta \in (0, 1)$, $\rho_k > 0$.

1. Set $\alpha = \Delta_k$
2. **While** $f(x_k \pm \alpha d_k) > f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$ **do**
 - If** $\alpha \|d_k\| < \rho_k$ **then**
 - set $\eta_k = \alpha$, $\alpha_k = 0$ and **exit**.
 - Else**
 - set $\alpha = \delta \alpha$.
 - End If****End while**
3. Let $u \in \{-1, 1\}$ be such that

$$f(x_k + u \alpha d_k) \leq f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$$

and set $\alpha = u \alpha$.

4. If $|\alpha| < \Delta_k$ set $\alpha_k = \alpha$ and **exit**.
5. **While**

$$f(x_k + \alpha d_k) < f(x_k) - \gamma_2 \alpha^2 \|d_k\|^2,$$

$$f(x_k + (\alpha/\delta) d_k) < \min \left\{ f(x_k + \alpha d_k), f(x_k) - \gamma_1 (\alpha/\delta)^2 \|d_k\|^2 \right\}$$

set $\alpha = \alpha/\delta$.

- End while**
6. Set $\alpha_k = \alpha$ and **exit**. □

It is easily seen that, if we assume that f is bounded below on R^n , the preceding algorithm is well defined.

In fact, both the while cycles at Step 2 and that at Step 5 terminate in a finite number of inner iterations.

The cycle at Step 2 terminates because $\alpha \leq \delta_u \alpha$ with $\delta_u < 1$ at each inner step and hence, at least the condition $\alpha \|d_k\| < \rho_k$ will be satisfied in a finite number of steps. The cycle at Step 5 terminates because, otherwise, we will have that $|\alpha| \rightarrow \infty$ and $f(x_k + \alpha d_k) \rightarrow -\infty$, which contradicts the assumption that f is bounded below.

Now, we can establish the convergence properties of Algorithm DFAGLS, under the same assumptions considered for the Armijo-type algorithm DFALS.

Proposition 10.7 (Convergence of Algorithm DFAGLS) Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that:

- (i) for every k , we have $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ and the step-size α_k is computed by employing Algorithm DFAGLS;
- (ii) we have $\rho_k \rightarrow 0$ for $k \rightarrow \infty$.

Then, we have

- (c₁) $x_k \in \mathcal{L}_0$ for all k ;
- (c₂) the sequences $\{f(x_k)\}$ converges to a limit;
- (c₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c₄) $\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$.

Proof Reasoning as in the proof of Proposition 10.6, we have that assertions (c₁), (c₂) and (c₃) must hold. Moreover, reasoning by contradiction, if we assume that (c₄) is false, we can find an infinite subsequence, which we relabel $\{x_k\}$, such that

$$\lim_{k \rightarrow \infty} x_k = \bar{x},$$

and

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} \neq 0, \quad \text{where } \|\bar{d}\| = 1. \quad (10.45)$$

Now, if we suppose that there exists \bar{k} such that, for all $k \geq \bar{k}$ (in the subsequence considered), the algorithm terminates at Step 2 or at Step 4, we can repeat the same reasonings followed in the proof of Proposition 10.6 and we get a contradiction to (10.45).

Therefore, we must assume that there exists an infinite subsequence (again relabeled $\{x_k\}$) such that $|\alpha| = \Delta_k$ before starting Step 5. Then, at termination of Step 5, because of the instructions of the algorithm, the step-size α_k must satisfy the condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \gamma_1 \alpha_k^2 \|d_k\|^2, \quad (10.46)$$

and at least one of the following conditions must be true.

$$f(x_k + \alpha_k d_k) \geq f(x_k) - \gamma_2 \alpha_k^2 \|d_k\|^2, \quad (10.47)$$

$$f(x_k + \frac{\alpha_k}{\delta} d_k) \geq f(x_k) - \gamma_1 \frac{\alpha_k^2}{\delta^2} \|d_k\|^2, \quad (10.48)$$

$$f(x_k + \frac{\alpha_k}{\delta} d_k) \geq f(x_k + \alpha_k d_k). \quad (10.49)$$

Using the Theorem of the Mean with reference to each pair of conditions (10.46)(10.47), (10.46)(10.48) or (10.46), (10.49), using (c₃) which implies $\|\alpha_k d_k\| \rightarrow 0$ and taking limits for $k \rightarrow \infty$, reasoning as in the proof of Proposition 10.6, we obtain $\nabla f(\hat{x})^T \hat{d} = 0$, which contradicts (10.45). This proves (c₄) and concludes the proof. \square

As we will see in the sequel, in some cases it could be of interest to perform derivative-free linesearches along directions d_k only for positive values of the tentative step-sizes. In this case we can modify Algorithm DFAGLS according to the following (conceptual) scheme.

Der-Free Armijo-Goldstein-Type Linesearch with Positive Tentative Steps (DFAGLSP)

Data. $\Delta_k > 0$, $\gamma_2 > \gamma_1 > 0$, $\delta \in (0, 1)$, $\rho_k > 0$.

1. Set $\alpha = \Delta_k$
2. **While** $f(x_k + \alpha d_k) > f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$ **do**
 - If** $\alpha \|d_k\| < \rho_k$ **then**
 - set $\eta_k = \alpha$, $\alpha_k = 0$ and **exit**.
 - Else**
 - set $\alpha = \delta \alpha$.
 - End If****End while**
3. If $\alpha < \Delta_k$ set $\alpha_k = \alpha$ and **exit**.
4. **While**

$$f(x_k + \alpha d_k) < f(x_k) - \gamma_2 \alpha^2 \|d_k\|^2,$$

$$f(x_k + (\alpha/\delta) d_k) < \min \left\{ f(x_k + \alpha d_k), f(x_k) - \gamma_1 (\alpha/\delta)^2 \|d_k\|^2 \right\}$$

$$\text{set } \alpha = \alpha/\delta.$$

End while

5. Set $\alpha_k = \alpha$ and **exit**. \square

Under the assumptions stated in Proposition 10.7, we can establish the convergence of the algorithm. In this case, however, we can easily prove that (c₁), (c₂) and (c₃)

of Proposition 10.7 still hold, but (c₄) has to be modified, as shown in the next proposition.

Proposition 10.8 (Convergence of Algorithm DFAGLSP) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that:*

- (i) *for every k , we have $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ and the step-size α_k is computed by employing Algorithm DFAGLSP;*
- (ii) *we have $\rho_k \rightarrow 0$ for $k \rightarrow \infty$.*

Then, we have

- (c₁) $x_k \in \mathcal{L}_0$ for all k ;
- (c₂) *the sequences $\{f(x_k)\}$ converges to a limit;*
- (c₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c₄) $\liminf_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| \geq 0$.

Proof Reasoning as in the proof of Proposition 10.6, we have again that assertions (c₁), (c₂) and (c₃) must hold. Now, suppose that (c₄) is false. Then, taking into account the compactness of \mathcal{L}_0 and assertion (c₁), we must find an infinite subsequence, which we relabel $\{x_k\}$, such that $x_k \rightarrow \bar{x}$ and

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} < 0, \quad \text{where } \|\bar{d}\| = 1. \quad (10.50)$$

Now, if the algorithm terminates at step 3 or at step 5, for some infinite subsequence, reasoning as in the proof of Proposition 10.8, we can establish that $\nabla f(\bar{x})^T \bar{d} = 0$, which contradicts (10.50).

Suppose now that in some infinite subsequence, say $\{x_k\}_K$, (of the subsequence considered in (10.50)), the algorithm terminates at step 2 with $\alpha_k = 0$. By the instructions of the algorithm this implies

$$f(x_k + \eta_k d_k) > f(x_k) - \gamma_1 \eta_k^2 \|d_k\|^2, \quad k \in K.$$

It can be easily verified, using the theorem of the mean and taking limits, that this implies $\nabla f(\bar{x})^T \bar{d} \geq 0$, which yields again a contradiction. Then (c₄) must hold. \square

Remark 10.2 If we assume that d_k is a descent direction, that is, if $\nabla f(x_k)^T d_k < 0$ for all k (and hence that the algorithm cannot terminate, in principle, for all sufficiently large k , with $\alpha_k = 0$), we will reobtain in the limit that $\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$. \square

10.6 Appendix A: Algorithms Employing Wolfe Conditions

We consider here the conceptual models of algorithms based on Wolfe conditions, which can reduce at each step an interval of uncertainty $[\alpha_l, \alpha_u]$, until a point satisfying the required conditions is determined. When needed, the interval is reduced by employing appropriate techniques of interpolation and extrapolation for generating a tentative point in the current interval.

In the sequel, when we need to show the dependence on the counter j of internal iterations of the line search, we indicate by $\alpha_l^{(j)}, \alpha_u^{(j)}$ the extreme points of the interval at the beginning of step j and by $\alpha^{(j)}$ the tentative point.

We consider two algorithms for satisfying weak or strong Wolfe conditions in a finite number of steps.

Weak Wolfe Conditions

The algorithm reported below determines a step-size α_k satisfying conditions (W1).

ALGW1: Line Search for Satisfying Conditions (W1)

Data. $\alpha_l = 0, \alpha_u = \infty$.

For $j = 0, 1, \dots$:

 Compute a tentative point $\alpha \in (\alpha_l, \alpha_u)$.

 If α satisfies conditions (W1) set $\alpha_k = \alpha$ and terminate.

 If $\phi(\alpha) > \phi(0) + \gamma\alpha\dot{\phi}(0)$ set $\alpha_u = \alpha$.

 If $\phi(\alpha) \leq \phi(0) + \gamma\alpha\dot{\phi}(0)$ and $\dot{\phi}(\alpha) < \sigma\dot{\phi}(0)$ set $\alpha_l = \alpha$.

End For

In the next proposition we show that the algorithm terminates, provided that the tentative point is not too close to the extreme points of the interval.

Proposition 10.9 (Convergence of ALGW1) *Suppose that the function $\phi : R^+ \rightarrow R$ is continuously differentiable on R^+ , bounded below and such that $\dot{\phi}(0) < 0$. Let $\theta \in [1/2, 1)$ and $\tau > 1$ be two given numbers and suppose that the tentative point α satisfies the following conditions for $j \geq 1$:*

- (i) $\alpha \geq \tau \max[\alpha_l, \alpha^{(0)}]$, if $\alpha_u = \infty$;
- (ii) $\max[(\alpha - \alpha_l), (\alpha_u - \alpha)] \leq \theta(\alpha_u - \alpha_l)$ if $\alpha_u < \infty$.

Then ALGW1 terminates and determines a step-length α_k satisfying the weak Wolfe conditions (10.34), (10.35).

Proof By contradiction, suppose that the assertion is false. First consider the case where $\alpha_u^{(j)} = \infty$ for all j . By assumption (i), for $j \geq 1$ we have that $\alpha^{(j)} \geq \tau^j \alpha^{(0)}$ and hence the instructions of the algorithm imply that $\alpha^{(j)} \rightarrow \infty$ for $j \rightarrow \infty$. On the other hand, as the condition $\phi(\alpha^{(j)}) \leq \phi(0) + \gamma \alpha^{(j)} \dot{\phi}(0)$ must be satisfied (for, otherwise, we should have $\alpha_u^{(j)} < \infty$), we have that $\phi(\alpha^{(j)}) \rightarrow -\infty$ for $j \rightarrow \infty$ and this contradicts the assumption that ϕ is bounded below.

Assume now that $\alpha^{(j)}_u \leq M$ for all sufficiently large j and some $M < \infty$, and that, for all j the tentative step-size does not satisfy (W1). Then the algorithm produces the two monotone bounded sequences $\{\alpha_l^{(j)}\}$ and $\{\alpha_u^{(j)}\}$ with $\alpha_l^{(j)} \geq 0$ and $\alpha_u^{(j)} \leq M$. Moreover, by assumption (ii), we have, for all sufficiently large j :

$$\alpha_u^{(j+1)} - \alpha_l^{(j+1)} \leq \theta (\alpha_u^{(j)} - \alpha_l^{(j)}),$$

which implies that $\alpha_u^{(j)} - \alpha_l^{(j)} \rightarrow 0$ for $j \rightarrow \infty$. It follows that both sequences converge to the same limit $\bar{\alpha}$ and hence that also $\alpha^{(j)} \in [\alpha_l^{(j)}, \alpha_u^{(j)}]$ converges to $\bar{\alpha}$. By the instruction of the algorithm, for all sufficiently large j , we can write:

$$\phi(\alpha_u^{(j)}) > \phi(0) + \gamma \alpha_u^{(j)} \dot{\phi}(0), \quad (10.51)$$

$$\phi(\alpha_l^{(j)}) \leq \phi(0) + \gamma \alpha_l^{(j)} \dot{\phi}(0). \quad (10.52)$$

Taking limits for $j \rightarrow \infty$ we obtain:

$$\phi(\bar{\alpha}) = \phi(0) + \gamma \bar{\alpha} \dot{\phi}(0), \quad (10.53)$$

and therefore, taking (10.51) into account, we have: $\alpha_u^{(j)} > \bar{\alpha}$. Moreover, by (10.53) and (10.51) we can write:

$$\begin{aligned} \phi(\alpha_u^{(j)}) &> \phi(0) + \gamma \alpha_u^{(j)} \dot{\phi}(0) \\ &= \phi(0) + \gamma (\bar{\alpha} + \alpha_u^{(j)} - \bar{\alpha}) \dot{\phi}(0) \\ &= \phi(\bar{\alpha}) + \gamma (\alpha_u^{(j)} - \bar{\alpha}) \dot{\phi}(0), \end{aligned} \quad (10.54)$$

whence it follows, for large j :

$$\frac{\phi(\alpha_u^{(j)}) - \phi(\bar{\alpha})}{\alpha_u^{(j)} - \bar{\alpha}} > \gamma \dot{\phi}(0).$$

Taking limits for $j \rightarrow \infty$ we have:

$$\dot{\phi}(\bar{\alpha}) \geq \gamma \dot{\phi}(0). \quad (10.55)$$

On the other hand, our assumptions imply that $\dot{\phi}(\alpha_l^{(j)}) < \sigma\dot{\phi}(0)$, and hence, taking limits, we get $\dot{\phi}(\bar{\alpha}) \leq \sigma\dot{\phi}(0)$. Therefore, as $\sigma > \gamma$ and $\dot{\phi}(0) < 0$ we get a contradiction with (10.55). \square

Strong Wolfe Conditions

Now we consider an algorithm that determines in a finite number of steps a point α_k satisfying the strong Wolfe conditions (W2).

To satisfy conditions (W2) we must modify algorithm ALGW1 in order to consider the case when the tentative point satisfies the weak but not the strong Wolfe conditions. This may occur when the slope condition (10.37) is violated at the tentative step-size α , because of the fact that $\dot{\phi}(\alpha) > \sigma|\dot{\phi}(0)|$. In this case we can update the upper bound by taking $\alpha_u = \alpha$. Then we can define the following scheme.

ALGW2: Line Search for Satisfying Conditions (W2)

Data. $\alpha_l = 0$, $\alpha_u = \infty$.

For $j = 0, 1, \dots$:

Determine a tentative point $\alpha \in (\alpha_l, \alpha_u)$.

If α satisfies conditions (W2) take $\alpha_k = \alpha$ and terminate.

If $\phi(\alpha) > \phi(0) + \gamma\alpha\dot{\phi}(0)$ set $\alpha_u = \alpha$.

If $\phi(\alpha) \leq \phi(0) + \gamma\alpha\dot{\phi}(0)$ and $\dot{\phi}(\alpha) < \sigma\dot{\phi}(0)$ set $\alpha_l = \alpha$.

If $\phi(\alpha) \leq \phi(0) + \gamma\alpha\dot{\phi}(0)$ and $\dot{\phi}(\alpha) > \sigma|\dot{\phi}(0)|$ set $\alpha_u = \alpha$.

End For

The convergence of algorithm ALGW2 can be established along the same lines followed for proving Proposition 10.9.

Proposition 10.10 (Convergence of ALGW2) Suppose that the function $\phi : R^+ \rightarrow R$ is continuously differentiable on R^+ , bounded below and such that $\dot{\phi}(0) < 0$. Let $\theta \in [1/2, 1]$ and $\tau > 1$ be two given numbers and suppose that the tentative point α satisfies the following conditions for $j \geq 1$:

- (i) $\alpha \geq \tau \max[\alpha_l, \alpha^{(0)}]$, if $\alpha_u = \infty$;
- (ii) $\max[(\alpha - \alpha_l), (\alpha_u - \alpha)] \leq \theta(\alpha_u - \alpha_l)$ if $\alpha_u < \infty$.

Then ALGW2 terminates and determines a step-length α_k satisfying the strong Wolfe conditions (10.36), (10.37).

Proof Reasoning by contradiction, as in the proof of Proposition 10.9, and using the same notation we can assert that $\alpha_u^{(j)} \leq M$ for all sufficiently large j and some

$M < \infty$, and that the sequences $\{\alpha_l^{(j)}\}$, $\{\alpha_u^{(j)}\}$ and $\{\alpha^{(j)}\}$ converge to the same limit $\bar{\alpha}$. As an infinite sequence of tentative points is generated, we can assume, without loss of generality that there exist two subsequences that we will indicate through the index sets J_1, J_2 such that at least one of these sets is infinite and we have

$$\phi(\alpha_u^{(j)}) > \phi(0) + \gamma \alpha_u^{(j)} \dot{\phi}(0), \quad j \in J_1, \quad (10.56)$$

$$\dot{\phi}(\alpha_u^{(j)}) > \sigma |\dot{\phi}(0)| \quad j \in J_2, \quad (10.57)$$

$$\phi(\alpha_l^{(j)}) \leq \phi(0) + \gamma \alpha_l^{(j)} \dot{\phi}(0) \quad j \in J_1 \cup J_2. \quad (10.58)$$

Moreover, by the instructions of the algorithm, we have

$$\dot{\phi}(\alpha_l^{(j)}) < \sigma \dot{\phi}(0) \quad j \in J_1 \cup J_2. \quad (10.59)$$

If J_1 is an infinite set we can repeat the same proof given for Proposition 10.9, with reference to the subsequence defined by $j \in J_1$ and we get a contradiction. Thus we can assume that J_1 is finite and that J_2 is infinite. Then, for sufficiently large $j \in J_2$ we have $\dot{\phi}(\alpha_u^{(j)}) > \sigma |\dot{\phi}(0)|$ so that, taking limits for $j \in J_2, j \rightarrow \infty$ we obtain $\dot{\phi}(\bar{\alpha}) \geq \sigma |\dot{\phi}(0)| > 0$. On the other hand, from (10.59), taking limits, we obtain $\dot{\phi}(\bar{\alpha}) \leq \sigma \dot{\phi}(0) < 0$ and this yields a contradiction. \square

10.7 Appendix B: Implementation of Line Searches

In this appendix we give some basic indications on the computer implementation of a line search procedure. Useful references on the realization of (monotone) computational line search schemes are the book [67] and the paper [187].

The line search has an important role in descent methods for unconstrained minimization. In fact all function evaluations and, possibly gradient evaluations, are performed during the line search and hence efficiency of this search has a great influence on the efficiency of the whole algorithm. However, a good implementation should be related to the specific choice of the search direction and often also to the class of problems under study. Moreover, numerical problems arising from finite precision of computer calculations should be carefully taken into account.

In this section we give only some general indication on the computer implementation of the line search algorithms described in this chapter. In particular, we add some comments on:

- the definition of an interval where the search must be carried out;
- the criteria for choosing the initial step-size Δ_k ;
- the use of interpolation techniques;
- the definition of diagnostic criteria and possible failures.

We remark, however, that concrete realization of line search codes typically contains many heuristic rules connected to the choice of the parameters and also possible refinements of the basic schemes.

We suppose here that the sequence $\{x_k\}$ is defined by $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction satisfying

$$\nabla f(x_k)^T d_k < 0$$

and we will indicate by $x_k(j)$ and $d_k(j)$ the j -th component of the n -vectors x_k , d_k .

10.7.1 Initial Interval

It is usually advisable to define an initial interval $[\alpha_{\min}, \alpha_{\max}]$ where the search must be carried out. In fact, because of the finite precision of computations, too small value of α could imply that x_{k+1} does not differ from x_k , while too large values of α typically indicate that the level set is unbounded.

As $x_{k+1} = x_k + \alpha_k d_k$, a reasonable indication is that α_k must be sufficiently large for determining an appreciable variation of at least one component of x_k . The largest (relative) magnitude of the components of d_k could be measured, for instance, by taking:

$$s_{\max} = \max_{1 \leq j \leq n} \frac{|d_k(j)|}{|x_k(j)| + 1}.$$

Then we could assume $\alpha_{\min} = \eta_m^{2/3}/s_{\max}$, where η_m is the *machine precision*. In double precision an indicative value could be

$$\alpha_{\min} \approx 10^{-11}/s_{\max}.$$

Using again the estimate s_{\max} introduced above we could set $\alpha_{\max} = M/s_{\max}$, where M is a sufficiently large value (e.g. $M = 10^3 \div 10^6$).

10.7.2 Initial Estimate of the Step-Size

The choice of the initial step-length $\Delta_k > 0$ has often a relevant effect on the number of function evaluations needed for determining α_k . We have already remarked that an important distinction is that between:

- methods where we must choose $\Delta_k = 1$;
- methods where d_k does not suggest an a priori choice for Δ_k .

In the latter case it is advisable to determine at least a plausible order of magnitude for Δ_k . One possible criterion is that of choosing Δ_k on the basis of a quadratic approximation of ϕ , of the form:

$$q(\alpha) = \lambda_0 + \lambda_1\alpha + \lambda_2\alpha^2.$$

We impose that $\lambda_0 = \phi(0)$, $\lambda_1 = \dot{\phi}(0)$ and that at the minimum point α^* the function q has a prescribed value f^* , which is the value we predict for ϕ at the end of the line search.

Recalling Proposition 10.1 we have:

$$q(\alpha^*) = \phi(0) + \frac{1}{2}\alpha^*\dot{\phi}(0), \quad (10.60)$$

so that, letting $q(\alpha^*) = f^*$, we can write:

$$\alpha^* = -2(\phi(0) - f^*)/\dot{\phi}(0).$$

If we indicate by $Df = f(x_k) - f^*$ the predicted reduction of f , we have:

$$\alpha^* = -\frac{2Df}{\nabla f(x_k)^T d_k}.$$

For $k = 0$ the value f^* and hence the predicted reduction Df should be estimated on the basis of the problem function. As default value, we could assume

$$Df = |f(x_0)| + 1$$

or some fraction of this value. For $k \geq 1$ we can assume as predicted reduction the actual reduction obtained in the last iteration, that is

$$Df = f(x_{k-1}) - f(x_k).$$

However, we must impose suitable safeguards on the estimated initial step-length; for instance we can assume:

$$\Delta_k = \min \left[-2 \frac{\max [Df, 10\eta_f]}{\nabla f(x_k)^T d_k}, \Delta_{\max} \right],$$

where η_f is the relative precision on the objective value and Δ_{\max} is an upper bound on Δ_k . As an example, we can assume $\eta_f = 10^{-6} \div 10^{-8}$, $\Delta_{\max} = 1$.

An alternative criterion could be that of choosing Δ_k in a way that the predicted first order variation of the objective function is equal to that observed in the

preceding iteration, that is:

$$\Delta_k = \frac{\alpha_{k-1} |\nabla f(x_{k-1})^T d_{k-1}|}{|\nabla f(x_k)^T d_k|}.$$

However, we must again check whether this value is numerically acceptable.

Another simple criterion, which can be used even if the gradient is not available, could be that of taking, for $k \geq 1$:

$$\Delta_k = \sigma \frac{\alpha_{k-1} \|d_{k-1}\|}{\|d_k\|},$$

with $0 < \sigma < 1$.

Also the magnitude of this step should be controlled and suitable safeguards should be introduced.

10.7.3 Interpolation Formulas

The interpolation formulas used in line searches are based on a quadratic or cubic model of the function $\phi(\alpha) = f(x_k + \alpha d_k)$.

10.7.3.1 Quadratic Interpolation

In quadratic interpolation we approximate $\phi(\alpha)$ with a strictly convex quadratic function $q(\alpha)$ of the form:

$$q(\alpha) = \lambda_0 + \lambda_1 \alpha + \lambda_2 \alpha^2,$$

where $\dot{q}(\alpha) = \lambda_1 + 2\lambda_2 \alpha$ and $\ddot{q}(\alpha) = 2\lambda_2$. We require that the minimizer α_q^* of q satisfies the conditions

$$\dot{q}(\alpha_q^*) = 0 \quad \ddot{q}(\alpha_q^*) > 0$$

so that we have

$$\alpha_q^* = -\frac{\lambda_1}{2\lambda_2}, \quad \lambda_2 > 0. \quad (10.61)$$

The parameters of q and hence the optimal step-size α_q^* can be determined on the basis of the available information. The cases of interest are given below; we indicate by α_1, α_2 two points of the real axis such that $\alpha_2 > \alpha_1$.

Case 1. At the points α_1, α_2 we know $\phi(\alpha_1), \phi(\alpha_2), \dot{\phi}(\alpha_1)$.

In this case we impose the conditions

$$q(\alpha_1) = \phi(\alpha_1), \quad q(\alpha_2) = \phi(\alpha_2) \quad \dot{q}(\alpha_1) = \dot{\phi}(\alpha_1) \quad (10.62)$$

In order to simplify our analysis, let us assume that $\alpha_1 = 0$, $\phi(\alpha_1) = \phi(0)$, $\alpha_2 = \alpha$, $\phi(\alpha_2) = \phi(\alpha)$, $\dot{\phi}(\alpha_1) = \dot{\phi}(0)$. This corresponds, typically, at the interpolation phase of Armijo-type methods. We can impose the conditions

$$q(0) = \phi(0), \quad q(\alpha) = \phi(\alpha) \quad \dot{q}(0) = \dot{\phi}(0). \quad (10.63)$$

Therefore, we have

$$\lambda_0 = \phi(0), \quad \lambda_1 = \dot{\phi}(0), \quad \lambda_2 = \frac{\phi(\alpha) - \phi(0) - \dot{\phi}(0)\alpha}{\alpha^2}.$$

If $\lambda_2 > 0$ we can compute the minimizer

$$\alpha_q^* = \left[\frac{-\dot{\phi}(0)\alpha^2}{2(\phi(\alpha) - \phi(0) - \alpha\dot{\phi}(0))} \right]. \quad (10.64)$$

We can obtain the general case by simply translating the interval of interest. By replacing 0 with α_1 , $\alpha_1 + \alpha$ with α_2 , $\phi(\alpha)$ with $\phi(\alpha_2)$ and α_q^* with $\hat{\alpha}_q = \alpha_1 + \alpha_q^*$ we obtain the minimizer

$$\hat{\alpha}_q = \alpha_1 - \frac{\dot{\phi}(\alpha_1)(\alpha_2 - \alpha_1)^2}{2(\phi(\alpha_2) - \phi(\alpha_1) - \dot{\phi}(\alpha_1)(\alpha_2 - \alpha_1))}, \quad (10.65)$$

provided that the second order derivative \ddot{q} given by

$$\ddot{q} = 2 \frac{\phi(\alpha_2) - \phi(\alpha_1) - \dot{\phi}(\alpha_1)(\alpha_2 - \alpha_1)}{(\alpha_2 - \alpha_1)^2}, \quad (10.66)$$

is positive.

Case 2. At the points α_1, α_2 we know $\dot{\phi}(\alpha_1), \dot{\phi}(\alpha_2)$.

In this case we can assume $\dot{q}(\alpha_1) = \dot{\phi}(\alpha_1)$, $\dot{q}(\alpha_2) = \dot{\phi}(\alpha_2)$ and we get

$$\alpha_q^* = \alpha_1 - \frac{\dot{\phi}(\alpha_1)(\alpha_2 - \alpha_1)}{\dot{\phi}(\alpha_2) - \dot{\phi}(\alpha_1)}, \quad (10.67)$$

provided that

$$\ddot{q} = (\dot{\phi}(\alpha_2) - \dot{\phi}(\alpha_1))/(\alpha_2 - \alpha_1) > 0.$$

Case 3. At the three points $\alpha_3 > \alpha_2 > \alpha_1$ we know the function values $\phi_1 = \phi(\alpha_1)$, $\phi_2 = \phi(\alpha_2)$, $\phi_3 = \phi(\alpha_3)$.

We can impose the conditions

$$q(\alpha_1) = \phi_1, \quad q(\alpha_2) = \phi_2, \quad q(\alpha_3) = \phi_3. \quad (10.68)$$

Then, solving the system in the unknown parameters $\lambda_0, \lambda_1, \lambda_2$ we obtain the minimizer

$$\alpha_q^* = \frac{1}{2} \frac{\phi_3(\alpha_2^2 - \alpha_1^2) + \phi_2(\alpha_1^2 - \alpha_3^2) + \phi_1(\alpha_3^2 - \alpha_2^2)}{\phi_3(\alpha_2 - \alpha_1) + \phi_2(\alpha_1 - \alpha_3) + \phi_1(\alpha_3 - \alpha_2)}, \quad (10.69)$$

provided that

$$\ddot{q} = 2 \frac{\phi_3(\alpha_2 - \alpha_1) + \phi_2(\alpha_1 - \alpha_3) + \phi_1(\alpha_3 - \alpha_2)}{(\alpha_3 - \alpha_2)(\alpha_3 - \alpha_1)(\alpha_2 - \alpha_1)} > 0.$$

10.7.3.2 Cubic Interpolation

In cubic interpolation we consider a third-order polynomial of the form

$$c(\alpha) = \lambda_0 + \lambda_1\alpha + \lambda_2\alpha^2 + \lambda_3\alpha^3,$$

Denoting by h the (constant) third-order derivative of c , we have:

$$\dot{c}(\alpha) = \lambda_1 + 2\lambda_2\alpha + 3\lambda_3\alpha^2, \quad \ddot{c}(\alpha) = 2\lambda_2 + 6\lambda_3\alpha, \quad h = 6\lambda_3.$$

Obviously the function is unbounded from below and we attempt to determine (if possible) a local minimizer α_c^* such that

$$\dot{c}(\alpha_c^*) = 0, \quad \ddot{c}(\alpha_c^*) > 0.$$

By employing Taylor's formula, we can write:

$$c(\alpha) = c(\alpha_1) + \dot{c}(\alpha_1)(\alpha - \alpha_1) + \frac{\ddot{c}(\alpha_1)}{2}(\alpha - \alpha_1)^2 + \frac{h}{6}(\alpha - \alpha_1)^3. \quad (10.70)$$

It follows

$$\dot{c}(\alpha) = \dot{c}(\alpha_1) + \ddot{c}(\alpha_1)(\alpha - \alpha_1) + \frac{h}{2}(\alpha - \alpha_1)^2, \quad (10.71)$$

$$\ddot{c}(\alpha) = \ddot{c}(\alpha_1) + h(\alpha - \alpha_1). \quad (10.72)$$

An important case is the following.

Case 4. At the points $\alpha_2 > \alpha_1 \geq 0$ we know

$$\phi_1 \equiv \phi(\alpha_1), \quad \dot{\phi}_1 \equiv \dot{\phi}(\alpha_1), \quad \phi_2 \equiv \phi(\alpha_2), \quad \dot{\phi}_2 \equiv \dot{\phi}(\alpha_2).$$

The interpolation conditions are

$$c(\alpha_1) = \phi_1 \quad \dot{c}(\alpha_1) = \dot{\phi}_1 \quad c(\alpha_2) = \phi_2, \quad \dot{c}(\alpha_2) = \dot{\phi}_2. \quad (10.73)$$

Then, using (10.70) (10.71) and imposing the preceding conditions we get:

$$\phi_2 = \phi_1 + \dot{\phi}_1(\alpha_2 - \alpha_1) + \frac{\ddot{c}(\alpha_1)}{2}(\alpha_2 - \alpha_1)^2 + \frac{h}{6}(\alpha_2 - \alpha_1)^3, \quad (10.74)$$

$$\dot{\phi}_2 = \dot{\phi}_1 + \ddot{c}(\alpha_1)(\alpha_2 - \alpha_1) + \frac{h}{2}(\alpha_2 - \alpha_1)^2. \quad (10.75)$$

Let us define the numbers

$$u = \frac{\ddot{c}(\alpha_1)}{2}, \quad v = \frac{h}{6}, \quad \eta = \alpha_2 - \alpha_1.$$

Then we obtain the following equations in the variables u and v .

$$\phi_2 - \phi_1 - \dot{\phi}_1 \eta = \eta^2 u + \eta^3 v, \quad (10.76)$$

$$\dot{\phi}_2 - \dot{\phi}_1 = 2\eta u + 3\eta^2 v. \quad (10.77)$$

Letting

$$s = \frac{\phi_2 - \phi_1}{\eta},$$

we obtain

$$u = \frac{1}{\eta} (3s - \dot{\phi}_2 - 2\dot{\phi}_1)$$

$$v = \frac{1}{\eta^2} (\dot{\phi}_1 + \dot{\phi}_2 - 2s).$$

Letting

$$\ddot{c}(\alpha_1) = 2u, \quad h = 6v \quad \xi = \alpha - \alpha_1$$

in (10.71) and imposing $\dot{c}(\alpha) = 0$, we obtain the second-degree equation

$$\phi_1 + 2u\xi + 3v\xi^2 = 0.$$

For $v \neq 0$ we obtain the solutions

$$\xi_{\pm}^{*} = \frac{-u \pm \sqrt{u^2 - 3\phi_1 v}}{3v}, \quad (10.78)$$

while, if $v = 0$ and $u \neq 0$ we have

$$\xi^{*} = -\frac{\phi_1}{2u}.$$

When $v \neq 0$, by evaluating the second order derivative, it can be shown that the minimizer is given by

$$\alpha_c^{*} = \alpha_1 + \frac{-u + \sqrt{u^2 - 3\phi_1 v}}{3v}.$$

When $u > 0$ a more stable equivalent form of this expression is given by:

$$\alpha_c^{*} = \alpha_1 + \frac{\phi_1}{-u - \sqrt{u^2 - 3\phi_1 v}},$$

which holds also when $v = 0$.

Another possible case is the following.

Case 5. At the points $\alpha_3 > \alpha_2 > \alpha_1$ we know $\phi(\alpha_1), \dot{\phi}(\alpha_1), \phi(\alpha_2), \phi(\alpha_3)$.

Now we can impose the interpolation conditions

$$c(\alpha_1) = \phi(\alpha_1) \quad \dot{c}(\alpha_1) = \dot{\phi}(\alpha_1) \quad c(\alpha_2) = \phi(\alpha_2), \quad c(\alpha_3) = \phi(\alpha_3). \quad (10.79)$$

Using Taylor's formula we can write

$$c(\alpha) = \phi(\alpha_1) + \dot{\phi}(\alpha_1)(\alpha - \alpha_1) + \frac{\ddot{c}(\alpha_1)}{2}(\alpha - \alpha_1)^2 + \frac{h}{6}(\alpha - \alpha_1)^3, \quad (10.80)$$

and we can determine the parameters $\ddot{c}(\alpha_1)$ and h by solving the system

$$c(\alpha_2) = \phi(\alpha_2),$$

$$c(\alpha_3) = \phi(\alpha_3).$$

Then we can determine α_c^{*} reasoning as in Case 4.

10.7.4 Application of Interpolation Formulas

We shortly discuss the application of the interpolation formulas in the algorithms considered in this chapter.

(a) *Armijo-type methods, with $\nabla f(x_k)$ given and function evaluations*

In backtracking Armijo-type methods we can assume $\alpha_1 = 0$ and interpolation is carried out in intervals of the form $[0, \alpha]$, where $\alpha > 0$ is the last tentative step-size where the acceptability condition was not satisfied.

If only the function value $\phi(\alpha)$ is computed at α we can refer to Case 1, with $\alpha_1 = 0$, $\dot{\phi}(0) = \nabla f(x_k) d_k < 0$ and $\alpha_2 = \alpha > 0$. Then we obtain

$$\ddot{q} = 2 \frac{\phi(\alpha) - \phi(0) - \dot{\phi}(0)\alpha}{\alpha^2}. \quad (10.81)$$

If $\ddot{q} \leq \varepsilon$, where $\varepsilon > 0$ is a small number, the approximation can not be accepted and we can use a reduction factor $\delta \in (0, 1)$, say, for instance, $\delta = 1/2$.

If $\ddot{q} > \varepsilon$, by (10.65), we can attempt to use the quadratic interpolation formula with suitable safeguards. We can write:

$$\alpha_q^* = \delta^* \alpha, \quad \text{where} \quad \delta^* = \frac{|\dot{\phi}(0)|\alpha}{2(\phi(\alpha) - \phi(0) + \alpha|\dot{\phi}(0)|)}$$

We require that the reduction factor δ used at the current iteration remains in a fixed interval, that is $\delta \in (l, u)$ where $0 < l < \delta < u$. We can take, for instance $l = 0.1, u = 0.9$. Then we can assume

$$\delta = \min [\max [\delta^*, l], u]$$

and the new step-size $\bar{\alpha}$ will be $\bar{\alpha} = \delta \alpha$. If $\bar{\alpha}$ is still not accepted, then we can repeat the safeguarded quadratic interpolation by replacing α and $\phi(\alpha)$ with $\bar{\alpha}$ and $\phi(\bar{\alpha})$.

Alternatively, we can perform a safeguarded cubic interpolation as in Case 5, using the last two values of ϕ and imposing that the new estimate $\bar{\alpha} = \delta \alpha$ solidifies the condition $\delta \in [l, u]$.

(b) *Armijo-Goldstein methods with $\nabla f(x_k)$ given and function evaluations*

In Armijo-Goldstein methods, which may require also step-length expansions, we can still use quadratic or cubic interpolations along the same lines followed in the preceding case. However, if a local minimizer has not been significantly bracketed and an extrapolation is required, we must check that the new tentative step-size satisfies a condition of the form $\bar{\alpha} \geq \tau \alpha$ with $\tau > 1$, (e.g. $\tau = 2 \div 5$).

(c) *Wolfe-type methods with function and gradient evaluations*

In this case we can use a quadratic (Case 2) or cubic interpolation (Case 4) or also an appropriate combination of the two estimates. In any case we must impose that the tentative step-size is not too close to the bounds of the uncertainty interval and that a significant variation of the step-length is obtained. If needed, we must obviously guarantee that a significant extrapolation is performed, as in the preceding case.

(d) *Interpolation based only on function values*

When we use only function evaluations we can adopt quadratic interpolation when three function values are available (Case 3). In particular, if $\alpha_1 = -\Delta$, $\alpha_2 = 0$ and $\alpha_3 = \Delta$ we get the quadratic estimate

$$\alpha_q^* = \frac{1}{2} \left(\frac{\phi(-\Delta) - \phi(\Delta)}{\phi(-\Delta) - 2\phi(0) + \phi(\Delta)} \right) \Delta, \quad (10.82)$$

provided that the denominator is positive. Then the step-length can be computed with the usual safeguards.

10.7.5 Stopping Criteria and Failures

During the line search, suitable tests must be performed in order to decide whether the search must be terminated, declaring a *failure* and indicating, when possible, the causes of this failure.

(a) *Tentative step-size $\alpha < \alpha_{\min}$ in Armijo-Goldstein methods*

When the tentative step-size is such that $\alpha < \alpha_{\min}$, but the condition of sufficient decrease has been not satisfied, the line search terminates with a failure. If α_{\min} has been defined appropriately, there are two possible explanations:

- the search direction d_k is not a descent direction; this may be due, for instance, to errors in the computation of f and/or ∇f or to an unappropriate definition of d_k ;
- the points produced by the algorithm can be trapped at the bottom of a steep valley and a reduction of f can not be obtained, because of the limited precision of computations.

(b) *Tentative step-size $\alpha > \alpha_{\max}$*

If $\alpha > \alpha_{\max}$ with a large value of α_{\max} this could indicate that the level sets are not bounded. In particular, this may happen when

$$\inf_{x \in R^n} f(x) = -\infty.$$

(c) *Too small search interval*

In the algorithms (such as Wolfe methods) where a search interval $[\alpha_l, \alpha_u]$ is reduced at each step, a line search failure can be observed when

$$\alpha_u - \alpha_l < \alpha_{\min}$$

and the causes can be essentially the same considered in case (a) in connection with Armijo-Goldstein methods.

10.8 Exercises

10.1 Show that Armijo-Goldstein method can be modified by terminating with $\alpha_k = \Delta_k$, provided that, when $\alpha = \Delta_k$ we have

$$f(x_k + \alpha d_k) \leq f(x_k) + \tilde{\gamma} \alpha \nabla f(x_k)^T d_k,$$

where $\tilde{\gamma} > 1$. Give a geometric interpretation of this situation.

10.2 Prove that, if Armijo-Goldstein line searches are used only at a subsequence $\{x_k\}_K$ of some algorithm $\{x_k\}$, then, if $f(x_{k+1}) \leq f(x_k)$ for all k , we can establish the limit

$$\lim_{k \in K, k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0.$$

10.3 Suppose that in Armijo's method the initial step-size is chosen in a way that

$$\frac{\rho_1}{\|d_k\|} \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \leq \Delta_k \leq \frac{\rho_2}{\|d_k\|} \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|},$$

where $\rho_2 \geq \rho_1 > 0$. Show that we have also

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

10.4 Prove Proposition 10.3 by replacing the Armijo condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k$$

with the following condition

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \gamma (\alpha_k \|d_k\|)^p,$$

where p is an integer greater or equal than 2.

10.9 Notes and References

Monotone inexact step-length algorithms are considered in most of introductory books on nonlinear programming. One of the first extensive study is given in [200]. An inexact one dimensional search (Goldstein conditions) has been first proposed in [117], Armijo's method has been introduced in [5] and Wolfe conditions have been defined in [257]. Additional useful results can be found in [12, 16, 67, 114, 196, 246]. Computational algorithms for the implementation of Wolfe conditions have been given in [187]. The derivative-free algorithms considered in this chapter are based on the papers [61] and [123]. Additional results concerning the *Fibonacci method* and the *Golden section method* can be found, for instance, in [12] and in [246]. Nonmonotone line search methods are studied in Chap. 24.

Chapter 11

Gradient Method



In this chapter we introduce the *gradient method*, which is one of the first methods proposed for the unconstrained minimization of differentiable functions and constitutes the prototype of a globally convergent algorithm. In particular, here we consider the standard monotone version, also known as *steepest descent method*, based on line searches, and a version with *constant step-sizes*. We prove global convergence and we report some estimates of the convergence rate. We show also that finite convergence can be attained, in principle, in the quadratic case, if the eigenvalues of the Hessian matrix are known. Finally, we indicate some *first order methods* proposed for modifying and improving the gradient method.

11.1 The Steepest Descent Direction

The *gradient method* for minimizing f makes use (when $\nabla f(x_k) \neq 0$) of the negative gradient direction at x_k for defining the search direction $d_k = -\nabla f(x_k)$. This choice is based on the fact that the (normalized) negative gradient direction is the vector that minimizes the directional derivative of f among the directions with unit Euclidean norm, and hence it is the solution of the problem:

$$\min \nabla f(x_k)^T d, \quad \|d\| = 1, \quad (11.1)$$

where $\|\cdot\|$ denotes the Euclidean norm. In fact, using Schwarz inequality, we can write $|\nabla f(x_k)^T d| \leq \|d\| \|\nabla f(x_k)\|$, where the equality sign holds if and only if we have $d = \lambda \nabla f(x_k)$, for some $\lambda \in \mathbb{R}$. Then, taking $\lambda = -1/\|\nabla f(x_k)\|$ the solution of (11.1) is given by: $d_k = -\nabla f(x_k)/\|\nabla f(x_k)\|$. By incorporating the scalar $1/\|\nabla f(x_k)\|$ into a step-size $\alpha_k > 0$, we can take $-\nabla f(x_k)$ as search

direction and we obtain the iteration

$$x_{k+1} = x_k + \alpha_k d_k \equiv x_k - \alpha_k \nabla f(x_k).$$

The choice of the search direction illustrated above motivates the term *steepest descent method*; we note however that the optimality of the solution of (11.1) depends on the particular choice of the norm.

The main interest of the negative gradient as search direction is essentially due to the fact that, when the gradient is continuous, the vector $-\nabla f(x)$ is a descent direction *continuous* in x , which is zero only when x is a stationary point. This allows us to establish easily a global convergence result.

11.2 The Gradient Method

We will refer to the following “conceptual” algorithm.

Algorithm 11.1 (Gradient Method)

1. Choose a starting point $x_0 \in R^n$.
- For $k=0,1,\dots$
2. Compute $\nabla f(x_k)$; if $\nabla f(x_k) = 0$ stop; otherwise set $d_k = -\nabla f(x_k)$.
3. Compute the step-size $\alpha_k > 0$ along d_k by means of a line search.
4. Set $x_{k+1} = x_k + \alpha_k d_k$.

End For

The next result is an immediate consequence of the results stated in the preceding chapters.

Proposition 11.1 (Convergence of the Gradient Method) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact.*

Let $\{x_k\}$ be the sequence generated by Algorithm 11.1, where the line search algorithm satisfies the properties

- (i) $f(x_{k+1}) < f(x_k)$ if $\nabla f(x_k) \neq 0$;
- (ii) if $\nabla f(x_k) \neq 0$ for all k , we have:

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

(continued)

Proposition 11.1 (continued)

Then, either there exists an index $v \geq 0$ such that $x_v \in \mathcal{L}_0$ and $\nabla f(x_v) = 0$, or else there exist limit points and every limit point of $\{x_k\}$ is a stationary point.

Proof As $d_k = -\nabla f(x_k)$ we have:

$$\frac{\nabla f(x_k)^T d_k}{\|d_k\|} = -\|\nabla f(x_k)\|. \quad (11.2)$$

Therefore, choosing the forcing function $\sigma(t) = t$ all the assumptions of Proposition 9.2 are satisfied and hence the assertion follows from this proposition. \square

On the basis of the preceding proposition we can construct a globally convergent version of the gradient method by employing, for instance, Armijo's line search with a fixed initial step-size, or a line search based on Wolfe conditions. We note, in particular, that a good choice of the initial step-size could be useful for reducing the computational cost and preventing *overflows* and hence the Armijo-Goldstein algorithm considered in the preceding chapter could be a convenient choice. Recalling the remarks already made on extensions of Armijo's method we can also easily establish the convergence of an exact line search along $-\nabla f(x_k)$.

The gradient method can be used, in association with other methods with better local convergence properties, in order to guarantee the global convergence. In particular, if the gradient method is used in correspondence to an infinite subsequence of a sequence generated with any descent method, under usual assumptions, we can ensure, at least, the existence of a subsequence converging to a stationary point. In this case, the gradient steps are often called *spacer steps*. In particular, we can establish the following result.

Proposition 11.2 Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence produced by the algorithm $x_{k+1} = x_k + \alpha_k d_k$ and suppose that

- (i) $f(x_{k+1}) \leq f(x_k)$ for all k ;
- (ii) there exists an infinite subsequence $\{x_k\}_K$ such that for $k \in K$ we choose the search direction $d_k = -\nabla f(x_k)$ and we compute the step-size α_k along $-\nabla f(x_k)$ by means, for instance, of an Armijo-type algorithm.

Then, if the algorithm does not terminate at a stationary point, the subsequence $\{x_k\}_K$ has limit points and every limit point is a stationary point of f .

Proof Let us denote by y_h the points of the subsequence by assuming $y_h = x_{k_h}$ for $k_h \in K$. By assumption we have: $f(y_{h+1}) \leq f(y_h + \alpha_{k_h} d_{k_h})$, where $d_{k_h} = -\nabla f(y_h)$ and α_{k_h} is the step-size computed with an Armijo-type line search. Then we have

$$\lim_{h \rightarrow \infty} \frac{\nabla f(y_h)^T d_{k_h}}{\|d_{k_h}\|} = 0$$

and the convergence properties of $\{y_h\}$ follow from Proposition 11.1 □

Note that in the preceding proposition we have not specified the choice of d_k and α_k for $k \notin K$; we have only assumed that $\{f(x_k)\}$ is non increasing.

11.3 Gradient Method with Constant Step-Size

Under suitable assumptions, we can show that convergence of the gradient method can be established even when we fix the step-size at a suitable constant value.

We need the following result, called *descent lemma* [16] where $d \in R^n$ is any given vector.

Proposition 11.3 Suppose that $f : R^n \rightarrow R$ is continuously differentiable over an open convex set D and let $x \in D$ and $d \in R^n$. Suppose that ∇f is Lipschitz continuous on D , that is that there exists $L > 0$ such that for all $w, u \in D$ we have

$$\|\nabla f(w) - \nabla f(u)\| \leq L\|w - u\|.$$

Let $\alpha \in R$ be such that $x + \alpha d \in D$. Then we have

$$f(x + \alpha d) \leq f(x) + \alpha \nabla f(x)^T d + \frac{\alpha^2 L}{2} \|d\|^2.$$

Proof Using the Theorem of the Mean we can write:

$$f(x + \alpha d) = f(x) + \alpha \int_0^1 \nabla f(x + t\alpha d)^T dt d,$$

which implies, taking into account the convexity of D and the Lipschitz-continuity of ∇f :

$$\begin{aligned} f(x + \alpha d) &= f(x) + \alpha \int_0^1 (\nabla f(x + t\alpha d)^T d - \nabla f(x)^T d) dt + \alpha \nabla f(x)^T d \\ &\leq f(x) + \alpha \int_0^1 \|\nabla f(x + t\alpha d) - \nabla f(x)\| \|d\| dt + \alpha \nabla f(x)^T d \\ &\leq f(x) + \alpha^2 L \int_0^1 t \|d\|^2 dt + \alpha \nabla f(x)^T d \\ &= f(x) + \frac{\alpha^2 L}{2} \|d\|^2 + \alpha \nabla f(x)^T d, \end{aligned}$$

which completes the proof. \square

Using this result we can establish convergence conditions for the gradient method with constant step-size.

Proposition 11.4 (Gradient Method with Constant Step-Size) Suppose that $f : R^n \rightarrow R$ is continuously differentiable on R^n and that the level set \mathcal{L}_0 is compact. Suppose further that ∇f is Lipschitz continuous on R^n , so that there exists $L > 0$ such that for all $w, u \in R^n$ we have

$$\|\nabla f(w) - \nabla f(u)\| \leq L\|w - u\|.$$

Let η be such that

$$\varepsilon \leq \eta \leq \frac{2 - \varepsilon}{L},$$

for some $1 > \varepsilon > 0$. Consider the sequence $\{x_k\}$ generated by the iterations

$$x_{k+1} = x_k - \eta \nabla f(x_k), \quad k = 0, 1, \dots$$

Then, either there exists some $v \geq 0$ such that $x_v \in \mathcal{L}_0$ and $\nabla f(x_v) = 0$, or else there exist limit points and every limit point of $\{x_k\}$ is a stationary point of f in \mathcal{L}_0 .

Proof Let $x_k \in \mathcal{L}_0$. Setting $d_k = -\nabla f(x_k)$ and using Proposition 11.3 we have:

$$f(x_{k+1}) = f(x_k + \eta d_k) \leq f(x_k) + \eta \nabla f(x_k)^T d_k + \frac{\eta^2 L}{2} \|d_k\|^2.$$

As $\nabla f(x_k)^T d_k = -\|\nabla f(x_k)\|^2$, it follows that

$$f(x_k) - f(x_{k+1}) \geq \eta \left(1 - \frac{\eta L}{2}\right) \|\nabla f(x_k)\|^2. \quad (11.3)$$

Thus, if $0 < \eta < \frac{2}{L}$, we have $f(x_{k+1}) < f(x_k)$, so that all points of the sequence $\{x_k\}$ remain in \mathcal{L}_0 . As $\{f(x_k)\}$ converges, we get in the limit:

$$\lim_{k \rightarrow \infty} \nabla f(x_k) = 0.$$

Then the assertion follows from known results. \square

The gradient method with constant step-size is not, in general, an efficient minimization technique and convergence cannot be guaranteed in absence of a reliable estimate of the Lipschitz constant. This method, however, has been used as an heuristic technique in the earliest papers on neural networks and is referred to as *backpropagation method*, because of the technique used for computing the gradient in multilayer networks [24].

11.4 Convergence Rate

The convergence rate of the gradient method has been studied under various assumptions on the objective function and on the choice of the step-size. We review here some of the best known results, where we refer to Euclidean norms. A standing assumption in all these analysis is the following.

Assumption 11.1 *The function $f : R^n \rightarrow R$ is twice continuously differentiable on R^n and there exist numbers $0 < m < M$ such that*

$$m\|u\|^2 \leq u^T \nabla^2 f(z)u \leq M\|u\|^2, \quad \text{for all } z, u \in R^n \quad (11.4)$$

The assumption stated implies that f is strongly convex on R^n , that the level set \mathcal{L}_0 is compact and that there exists a unique minimum point x^* .

Recalling the variational characterization of eigenvalues, we have that Assumption 11.1 is satisfied if

$$0 < m \leq \lambda_{\min}(\nabla^2 f(z)) \leq \lambda_{\max}(\nabla^2 f(z)) \leq M \quad \text{for all } z \in R^n,$$

where $\lambda_{\min}(\nabla^2 f(z))$ and $\lambda_{\max}(\nabla^2 f(z))$ denote, respectively, the minimum and the maximum eigenvalue of $\nabla^2 f(z)$.

When Assumption 11.1 is satisfied it is also easily seen that ∇f is Lipschitz continuous with Lipschitz constant M . In fact, given x, y in R^n we can write:

$$\nabla f(y) = \nabla f(x) + \int_0^1 \nabla^2 f(x + t(y - x))(y - x) dt, \quad (11.5)$$

whence it follows

$$\|\nabla f(y) - \nabla f(x)\| \leq \max_{0 \leq t \leq 1} \|\nabla^2 f(x + t(y - x))\| \|y - x\|.$$

On the other hand, recalling the expression of the matrix norm of a positive definite symmetric matrix induced by the Euclidean norm, and using Assumption 11.1, we have, for all $t \in [0, 1]$ and all $x, y \in R^n$:

$$\|\nabla^2 f(x + t(y - x))\| = \lambda_{\max}(\nabla^2 f(x + t(y - x))) \leq M,$$

so that

$$\|\nabla f(y) - \nabla f(x)\| \leq M\|y - x\| \quad \text{for all } x, y \in R^n. \quad (11.6)$$

Now we consider two special cases, where we get simple estimates of the convergence rate. In particular, we suppose first that f is strongly convex on R^n and that the step-size is constant; then we consider the quadratic case with positive definite Hessian matrix, under the assumption that the line search is performed using the optimal step-size.

When the step-size is constant for all k we have the following result.

Proposition 11.5 (Convergence Rate with Constant Step-Size) Suppose that $f : R^n \rightarrow R$ is twice continuously differentiable on R^n and that Assumption 11.1 is satisfied. Suppose that the sequence $\{x_k\}$ is generated, starting from a given point $x_0 \in R^n$, through the iterations

$$x_{k+1} = x_k - \eta \nabla f(x_k), \quad k = 0, 1, \dots,$$

where

$$0 < \eta < \frac{2}{M}$$

and that $\nabla f(x_k) \neq 0$ for all k . Then the sequence $\{x_k\}$ converges to the unique minimum point x^* with (at least) Q -linear convergence rate and we have

$$\|x_{k+1} - x^*\| \leq q \|x_k - x^*\|, \quad (11.7)$$

(continued)

Proposition 11.5 (continued)
where

$$q = \max \{ |1 - \eta m|, |1 - \eta M| \}.$$

Moreover, the minimum value q^* of q is

$$q^* = \frac{M - m}{M + m}, \quad (11.8)$$

and this value is reached when the step-size is

$$\eta^* = \frac{2}{M + m}. \quad (11.9)$$

Proof As discussed above, by Assumption 11.1 the hypotheses of Proposition 11.4 are satisfied and hence the sequence $\{x_k\}$ converges to the unique minimizer x^* , where $\nabla f(x^*) = 0$. Recalling (11.5) and letting

$$H_k = \int_0^1 \nabla^2 f(x^* + t(x_k - x^*)) \ dt,$$

as $\nabla f(x^*) = 0$ we have: $\nabla f(x_k) = \nabla f(x_k) - \nabla f(x^*) = H_k(x_k - x^*)$.

Then we can write

$$\begin{aligned} x_{k+1} - x^* &= x_k - x^* - \eta H_k(x_k - x^*) \\ &= (I - \eta H_k)(x_k - x^*). \end{aligned}$$

It follows

$$\|x_{k+1} - x^*\| \leq \|I - \eta H_k\| \|x_k - x^*\|. \quad (11.10)$$

Now, the induced Euclidean norm of the symmetric matrix $I - \eta H_k$ in terms of its eigenvalues, which we denote by σ_i^k , is given by

$$\|I - \eta H_k\| = \max_{1 \leq i \leq n} |\sigma_i^k| = \max_{1 \leq i \leq n} |1 - \eta \lambda_i^k|,$$

where λ_i^k are the eigenvalues of H_k . By Assumption 11.1 we have that $0 < m \leq \lambda_i^k \leq M$, and hence it can be easily verified that $\|I - \eta H_k\| \leq \max \{ |1 - \eta m|, |1 - \eta M| \}$, so that, recalling (11.10) the first assertion is proved.

In order to determine the value η^* in $(0, 2/M)$ that minimizes the function $q(\eta) = \max \{ |1 - \eta m|, |1 - \eta M| \}$ we can observe that the linear function $1 - \eta M$

vanishes and changes sign at $1/M$, whereas the linear function $1 - \eta m$ vanishes at $1/m$, which is to the right of $1/M$. Then the minimum value of the max function $q(\eta)$ is reached when

$$1 - \eta m = -(1 - \eta M),$$

so that $\eta^* = 2/(M + m)$, which yields the minimum value (11.8). \square

Consider now the case where the objective function is a strictly convex quadratic function. To simplify notation we assume $f(x) = 1/2x^T Qx$, where Q is a positive definite symmetric matrix. We assume that the step-size along the negative gradient is the optimal step-size, that is $\alpha_k = -\nabla f(x_k)^T d_k / d_k^T Q d_k$. Let us define the norm

$$\|x\|_Q = (x^T Q x)^{1/2}.$$

Then, we can state the following result, whose proof can be found, for instance, in [16].

Proposition 11.6 (Convergence Rate in the Quadratic Case) *The gradient method with optimal step-size converges to the global minimizer $x^* = 0$ of the function $f(x) = 1/2x^T Qx$, with Q symmetric positive definite and we have:*

$$\|x_{k+1} - x^*\|_Q \leq \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m} \right) \|x_k - x^*\|_Q, \quad (11.11)$$

where λ_M and λ_m are, respectively, the maximum and the minimum eigenvalue of Q . \square

As a consequence of the preceding proposition we can write, in the Euclidean norm:

$$\|x_k - x^*\| \leq C \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m} \right)^k, \quad (11.12)$$

where

$$C = \left(\frac{\lambda_M}{\lambda_m} \right)^{1/2} \|x_0 - x^*\|.$$

The results given above show essentially that the gradient method has at least a Q -linear convergence rate. In the quadratic case, the convergence rate depends on the

ratio $r = \lambda_M/\lambda_m$, which represents the *condition number* of the Hessian matrix. If $r = 1$, then convergence is attained in one step, while if $r > 1$, there are starting points such that the convergence speed greatly deteriorates for large values of r . These results are also indicative of the local behavior of the gradient method in the non quadratic case.

Whenever possible, a useful remedy to ill-conditioning problems can be that of preconditioning the gradient direction by pre-multiplying ∇f with some approximation to the inverse Hessian matrix.

11.5 Finite Convergence in the Quadratic Case

We show here that the gradient method terminates finitely at the minimizer of a strictly convex quadratic function defined on R^n , provided that the step-size are, in sequence, the inverse of the eigenvalues of the Hessian matrix.

Proposition 11.7 *Let $f(x) = \frac{1}{2}x^T Qx + c^T x$, and suppose that Q is positive definite, with eigenvalues λ_i , $i = 1, n$. Then the algorithm defined by*

$$x_{k+1} = x_k - \frac{1}{\lambda_k} \nabla f(x_k), \quad (11.13)$$

for $k = 1, \dots, n$, terminates in at most n steps at the minimum point x^ of f .*

Proof By (11.13) we can write:

$$\nabla f(x_{k+1}) \equiv Qx_{k+1} + c = Qx_k + c - \frac{1}{\lambda_k} Q \nabla f(x_k) = \left(I - \frac{1}{\lambda_k} Q \right) \nabla f(x_k).$$

Repeated application of this formula yields

$$\nabla f(x_k) = \prod_{j=1}^{k-1} \left(I - \frac{1}{\lambda_j} Q \right) \nabla f(x_1).$$

Let now $\{u_h \in R^n, h = 1, \dots, n\}$ be a set of n real linearly independent eigenvectors of Q , associated to the eigenvalues λ_h , $h = 1, \dots, n$. Then we have

$$Qu_h = \lambda_h u_h, \quad h = 1, \dots, n,$$

so that

$$\left(I - \frac{1}{\lambda_j} Q \right) u_h = (1 - \frac{\lambda_h}{\lambda_j}) u_h.$$

Taking $\{u_h \in R^n, h = 1, \dots, n\}$ as a basis in R^n , we can represent $\nabla f(x_1)$ in the form

$$\nabla f(x_1) = \sum_{h=1}^n \beta_h u_h,$$

where $\beta_h \in R$ are suitable scalars. Then, for $k \geq 2$ we can write:

$$\nabla f(x_k) = \left[\prod_{j=1}^{k-1} \left(I - \frac{1}{\lambda_j} Q \right) \right] \sum_{h=1}^n \beta_h u_h,$$

whence it follows for $k = n + 1$:

$$\nabla f(x_{n+1}) = \sum_{h=1}^n \beta_h \left[\prod_{j=1}^n \left(1 - \frac{1}{\lambda_j} \lambda_h \right) \right] u_h = 0,$$

which concludes our proof. \square

This result cannot be easily used, in practice, as the eigenvalues of the Hessian matrix are not available; however it yields useful indications in the study of new non monotone versions of the gradient method, mentioned in the next section and considered in Chap. 25, where the step-sizes can be related to suitable approximations of the inverse eigenvalues.

11.6 Complexity of the Steepest Descent Method

In this section, we study the complexity analysis of the steepest descent method with reference to the minimization of strictly convex quadratic functions and to the minimization of nonconvex smooth functions. First let us consider the following problem

$$\min f(x) = \frac{1}{2} x^T Q x + c^T x, \quad (11.14)$$

where Q is a $n \times n$ symmetric positive definite matrix. We study the iteration performance of the *steepest descent* method with exact line search. In particular, we show that the upper complexity estimate is $O\left(\log \frac{1}{\epsilon}\right)$.

According to the notation introduced in Chap. 8, we have that:

- \mathcal{C} is the class of unconstrained minimization problems of strictly convex quadratic functions;
- $\mathcal{O}(x) = \{f(x), \nabla f(x)\}$, that is, the oracle provides first order information;
- given $x_0 \in \mathbb{R}^n$, τ_ϵ is defined by $f(x) - f(x^*) \leq \epsilon(f(x_0) - f(x^*))$, with $\epsilon \in (0, 1)$, where x^* is the solution of the problem.

The iteration of the steepest descent method with exact line search is defined as follows

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k),$$

where

$$\alpha_k = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T Q \nabla f(x_k)}.$$

For simplicity we assume that f is an homogeneous convex quadratic function, i.e.,

$$f(x) = \frac{1}{2} x^T Q x,$$

so that the unique solution is $x^* = 0$ with $f(x^*) = 0$. From the proof (not reported in the book) of Proposition 11.6 it follows

$$f(x_{k+1}) \leq \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m} \right)^2 f(x_k),$$

where λ_M and λ_m are, respectively, the maximum and the minimum eigenvalue of Q . Setting $C = \frac{\lambda_M}{\lambda_m}$ we can write

$$f(x_{k+1}) \leq \left(\frac{C-1}{C+1} \right)^2 f(x_k).$$

Thus, recursively we obtain

$$\frac{f(x_k)}{f(x_0)} \leq \left(\frac{C-1}{C+1} \right)^{\frac{2kC}{C}},$$

from which it follows

$$\log \left(\frac{f(x_k)}{f(x_0)} \right) \leq \frac{2k}{C} \log \left(\frac{C-1}{C+1} \right)^C \quad (11.15)$$

Consider the following one dimensional function $h : [1, \infty) \rightarrow R$

$$h(t) = \left(\frac{t-1}{t+1} \right)^t$$

It can be easily verified that, for $t > 1$, $h(t)$ is an increasing function, furthermore, we have

$$\lim_{t \rightarrow \infty} \left(\frac{t-1}{t+1} \right)^t = \frac{1}{e^2}.$$

Using (11.15), it follows

$$\log \left(\frac{f(x_k)}{f(x_0)} \right) \leq \frac{2k}{C} \log \left(\frac{1}{e^2} \right) = \frac{-4k}{C}.$$

Let K be the set of the first iterations such that the stopping criterion

$$f(x_k) \leq \epsilon f(x_0)$$

is not satisfied, i.e.,

$$\frac{f(x_k)}{f(x_0)} > \epsilon \quad \forall k = 1, 2, \dots, |K|,$$

and

$$\frac{f(x_{|K|+1})}{f(x_0)} \leq \epsilon.$$

Therefore, we can write

$$\log \epsilon < \log \left(\frac{f(x_{|K|})}{f(x_0)} \right) \leq \frac{-4|K|}{C},$$

and we can conclude that

$$|K| \leq \frac{C}{4} \log \frac{1}{\epsilon}, \quad (11.16)$$

i.e., the upper bound on the number of iterations necessary to satisfy the stopping criterion is $O(\log \frac{1}{\epsilon})$.

Now consider the class of unconstrained optimization problems where:

- the objective function f is continuously differentiable;
- the gradient ∇f is Lipschitz continuous, that is, there exists a positive constant L such that, for all $x, y \in R^n$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

We assume that the objective function is bounded below and we denote by f^* a lower bound of f . As stopping criterion, we adopt

$$\|\nabla f(x)\| \leq \epsilon. \quad (11.17)$$

For simplicity we consider the *steepest descent* method with exact line search, that is, the method described as follows

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \quad (11.18)$$

where the stepsize α_k is

$$\alpha_k \in \arg \min_{\alpha \geq 0} f(x_k - \alpha \nabla f(x_k)).$$

We can state the following proposition.

Proposition 11.8 (Upper Complexity Estimate) *The number of iterations of the steepest descent method (11.18) to satisfy the stopping rule (11.17) is $O(\epsilon^{-2})$.*

Proof By the descent lemma we have

$$\begin{aligned} f(x_{k+1}) &= f(x_k - \alpha_k \nabla f(x_k)) = \min_{\alpha \geq 0} f(x_k - \alpha \nabla f(x_k)) \\ &\leq \min_{\alpha \geq 0} \{f(x_k) - \alpha \|\nabla f(x_k)\|^2 + \frac{1}{2}(\alpha)^2 L \|\nabla f(x_k)\|^2\} \\ &= f(x_k) + \|\nabla f(x_k)\|^2 \min_{\alpha \geq 0} \{-\alpha + \frac{1}{2}\alpha^2 L\} \\ &= f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2, \end{aligned}$$

from which we obtain

$$\frac{1}{2L} \sum_{i=0}^k \|\nabla f(x_i)\|^2 \leq f(x_0) - f(x_{k+1}) \leq f(x_0) - f^*.$$

Let K be the set of first iterations such that $\|\nabla f(x_i)\| > \epsilon$ for $i \in K$. We can write

$$\frac{1}{2L} \sum_{i \in K} \epsilon^2 \leq \frac{1}{2L} \sum_{i \in K} \|\nabla f(x_i)\|^2 \leq f(x_0) - f^*,$$

which implies

$$|K| \leq \frac{2L(f(x_0) - f^*)}{\epsilon^2},$$

i.e., the upper bound on the number of iterations necessary to satisfy the stopping criterion (11.17) is $O(\epsilon^{-2})$. \square

The same upper bound $O(\epsilon^{-2})$ on the number of iterations required to obtain an approximate stationary point was given by several variants of the steepest descent algorithm where the step-size is computed using the knowledge of the Lipschitz constant L , or by inexact line searches. Through one-dimensional counter-examples [39] it is shown that $O(\epsilon^{-2})$ is a lower bound on the number of iterations for all the monotone variants of the steepest descent method.

The same complexity bound $O(\epsilon^{-2})$ has been proved for gradient-related methods using a non-monotone linesearch [43]. However, the construction of an example illustrating the sharpness of this bound has not been provided and the difficulty in the construction of examples is due to the nonmonotonicity.

A comparison in terms of complexity analysis between the gradient method and methods using second order information will be discussed later.

11.7 Modified Gradient Methods

In order to improve the convergence properties of the standard gradient method, various *first order* techniques have been proposed, which make use only of first order derivatives, as the steepest descent method, and yet can guarantee considerable advantages in terms of convergence speed and robustness.

The modifications we consider here are essentially *two-step methods* where we make use also of information obtained in the preceding step (typically x_{k-1} and $\nabla f(x_{k-1})$) for computing the search direction at step k .

An interesting modification is the so called *Barzilai-Borwein gradient method* [11] or *spectral gradient method*, which determines a scaling coefficient of the negative gradient direction, on the basis of the information obtained at the preceding

step. This method, which may require the use of the non monotone line search techniques described in Chap. 24, has proved to be quite efficient, in practice, in large scale and ill-conditioned problems. An introduction to this method is given in Chap. 25.

A quite different approach, which is at the basis of various and important first order methods, is again a two-step method, which exploits information from the past iterate, by adding a *momentum term* $\beta(x_k - x_{k-1})$ to the negative gradient direction, so that, for $k \geq 1$, we have an iteration of the form:

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}), \quad (11.19)$$

where $\alpha > 0$ and $\beta \geq 0$ are suitable parameters, possibly dependent on k .

An algorithm of this form is the so-called *heavy ball method* [208]. This model is suggested by the physical analogy with the motion of a “ball” in a potential field in presence of a friction force that mitigates the “zigzagging” near the point with minimal value of the potential f . By discretizing the second order differential equation that describes the motion of the ball we obtain the discrete dynamical system described by (11.19). Under suitable differentiability and convexity assumptions, it can be shown that the method has a faster convergence rate in comparison with the gradient method without “memory”.

More specifically, suppose, for simplicity, that f is a quadratic function of the form

$$f(x) = \frac{1}{2}(x - x^*)^T Q(x - x^*),$$

with Q symmetric and positive definite and let $0 < \lambda_m \leq \lambda_M$ be the extreme eigenvalues of the Hessian matrix Q .

Then, if we assume $0 < \beta < 1$, $0 < \alpha < 2(1 + \beta)/\lambda_M$, it can be shown (see [208]) that the sequence $\{x_k\}$ generated by the heavy ball method converges to x^* . Moreover, if we take

$$\alpha^* = \frac{4}{(\sqrt{\lambda_M} + \sqrt{\lambda_m})^2} \quad \beta^* = \left(\frac{\sqrt{\lambda_M} - \sqrt{\lambda_m}}{\sqrt{\lambda_M} + \sqrt{\lambda_m}} \right)^2, \quad (11.20)$$

we have that

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} \leq \frac{\sqrt{\lambda_M} - \sqrt{\lambda_m}}{\sqrt{\lambda_M} + \sqrt{\lambda_m}}. \quad (11.21)$$

Thus, for large value of the condition number $\kappa = \lambda_M/\lambda_m$ we have

$$\frac{\sqrt{\lambda_M} - \sqrt{\lambda_m}}{\sqrt{\lambda_M} + \sqrt{\lambda_m}} \approx 1 - 2/\sqrt{\kappa},$$

When the condition number is large, this shows that the heavy ball method achieves a much faster convergence rate in comparison with the steepest descent direction, for which we have

$$\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m} \approx 1 - 2/\kappa.$$

The heavy ball method has been rediscovered and largely employed, on an heuristic basis, in the construction of learning algorithms for neural networks (see, for instance [24] and the references quoted there).

An iteration of the form (11.19) is also at the basis of the *Conjugate Gradient Method* (CGM), which belongs to the class of *conjugate directions methods*, originally introduced as iterative methods for solving linear systems [143] and later extended to the minimization of non quadratic functions [142] and [96]. Conjugate direction methods and further references are considered in Chap. 12. Here we only note that in the CGM the parameters α and β must be chosen appropriately at each step k , in a way that in the strictly convex quadratic case finite convergence is obtained in n steps. It is also possible to show that under strong convexity and differentiability assumptions the convergence speed depends on the square root of the condition number, as in the heavy ball method.

Another interesting modification of the gradient method has been suggested by constructing algorithms where gradient steps are alternated with suitable *acceleration steps* along a line connecting two points generated during the preceding iteration.

A method with this structure is the *gradient PARTAN (PARallel TANgent method)* proposed in [238], as a technique that minimizes a strictly convex quadratic function in a finite number of iteration.

As the CGM, it can be employed also for non quadratic functions by performing appropriate line searches along the search directions.

More specifically, following the description given in [177], suppose that $x_0 \in R^n$ is a given point and that we generate the points in R^n $y_0, x_1, y_1, x_2, y_2, \dots, y_{k-1}, x_k, \dots$ by taking

$$y_0 = x_0 - \alpha_0 \nabla f(x_0), \quad x_1 = y_0$$

and

$$y_k = x_k - \alpha_k \nabla f(x_k), \quad k = 1, 2, \dots, n, \dots$$

$$x_{k+1} = y_k + \beta_k(y_k - x_{k-1}), \quad k = 1, 2, \dots, n-1, \dots$$

In the quadratic case, if we perform exact linesearches along each direction, it can be shown that the minimizer is reached at x_n . Actually, the vectors $w_k = x_{k+1} - x_k$ for $k = 0, 1, \dots, n-1$ are the same directions that would be generated by the CGM, starting from the same initial point. In addition, it must be noted that, using Armijo-

type inexact line searches (for computing both α_k and β_k) the global convergence can be ensured in the general case, under the same assumptions stated for the steepest descent method. In fact, the gradient steps can be viewed as spacer steps and global convergence would follow from Proposition 11.2. However, in comparison with the CGM a possible disadvantage could be the need of performing more line searches, but the implementation in the general case has not been much investigated in recent years.

A more recent technique, based on a similar structure, is the *Nesterov's accelerated gradient method* proposed in [194]. In this work it is assumed that the objective function is convex and continuously differentiable with Lipschitz continuous gradient. It is shown that the iteration complexity is $O(1/\sqrt{\epsilon})$, which significantly improves the $O(1/\epsilon)$ complexity bound of the gradient descent method. However, to our knowledge, the practical implementation, the computational evaluation and the extension to general continuously differentiable functions have not been deeply investigated. Recent works concern combinations of the different approaches mentioned before and we may expect various efficient new proposals.

11.8 Exercises

11.1 Solve the problem

$$\begin{aligned} \min \quad & \nabla f(x_k)^T d, \\ & \|d\|_2 = 1 \end{aligned}$$

by employing the KKT conditions.

11.2 Prove that under Assumption 11.1 the level set is compact and there exists a unique minimum point of f on R^n .

11.3 Construct a computer code based on the gradient method, employing the Armijo-Goldstein line search, and perform numerical experiments with some test problems.

11.4 Consider a gradient method such that, for each $k \in K$, being $K \subset \{0, 1, \dots\}$,

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k),$$

where α_k is computed by Armijo's line search. Define $K = \{k_1, k_2, \dots, k_i, \dots\}$ and assume that there exists an integer $M > 0$ such that, for $i = 1, 2, \dots$,

$$k_{i+1} - k_i \leq M.$$

Suppose that for every $k \notin K$ we have

$$f(x_{k+1}) \leq f(x_k) - \sigma(\|x_{k+1} - x_k\|),$$

where $\sigma : R^+ \rightarrow R^+$ is a forcing function. Assume that the level set \mathcal{L}_0 is compact. Prove that the method admits limit points and that every limit point is a stationary point.

11.9 Notes and References

The gradient method is the most popular optimization method and is described and analyzed in all the nonlinear optimization books. An historical survey of the steepest descent method can be found in [203]. Gradient methods with momentum are deeply analyzed and investigated in [208] and [259].

Chapter 12

Conjugate Direction Methods



We consider *conjugate direction methods*, a class of algorithms originally introduced as iterative methods for solving linear systems with positive definite coefficient matrix (and hence for minimizing quadratic functions with positive definite Hessian matrix). In particular, after illustrating the essential concepts related to the conjugate direction method, we describe the *Conjugate Gradient Method* (CGM), which is the best known algorithm in this class, we give a short summary of the properties of this algorithm and we describe the extension to the minimization of non quadratic functions.

In the sequel, in order to simplify notation we indicate the gradient $\nabla f(x)$ with $g(x)$ whenever convenient; in particular $\nabla f(x_k)$ will be indicated by g_k .

12.1 The Conjugate Direction Method

Consider the quadratic function on R^n defined by

$$f(x) = 1/2x^T Qx - c^T x,$$

where Q is a symmetric $n \times n$ matrix and $c \in R^n$. We know that f has a minimum point if and only if Q is positive semidefinite and there exists a point x^* such that $\nabla f(x^*) = Qx^* - c = 0$. When the Hessian matrix Q is symmetric positive definite the function f is strictly convex and has a unique minimum point $x^* = Q^{-1}c$, which is the solution of the linear system $Qx = c$. We introduce the following definition.

Definition 12.1 (Conjugate Directions) Given a symmetric $n \times n$ matrix Q , two non zero vectors $d_i, d_j \in R^n$ are said to be conjugate with respect to Q (or Q -conjugate, or Q -orthogonal) if: $d_i^T Q d_j = 0$. \square

In the sequel, unless otherwise stated, we will assume that Q is also positive definite. When $Q = I$ the definition above reduces to the definition of orthogonal vectors.

Now we show that mutually conjugate vectors are linearly independent.

Proposition 12.1 (Linear Independence of Conjugate Directions) Let $d_0, d_1, \dots, d_m \in R^n$ be a set of non zero mutually conjugate vectors with respect to a $n \times n$ matrix Q symmetric and positive definite. Then d_0, d_1, \dots, d_m are linearly independent.

Proof Let $\alpha_0, \alpha_1, \dots, \alpha_m$ be real numbers such that $\sum_{j=0}^m \alpha_j d_j = 0$. Pre-multiplying by $d_i^T Q$, as $d_i^T Q d_j = 0$ for $i \neq j$, we obtain: $0 = \sum_{j=0}^m \alpha_j d_i^T Q d_j = \alpha_i d_i^T Q d_i$. On the other hand, as Q is positive definite and $d_i \neq 0$ we have $d_i^T Q d_i > 0$, so that we have necessarily $\alpha_i = 0$. Repeating the same reasoning for $i = 0, 1, \dots, m$ we can assert that $\alpha_i = 0$ for $i = 0, 1, \dots, m$, which proves the thesis. \square

In order to illustrate the advantages of the Q -conjugacy relation, suppose we have a set of n mutually Q -conjugate directions $\mathcal{D} = \{d_0, d_1, \dots, d_{n-1}\}$. By Proposition 12.1, \mathcal{D} is a linearly independent set, which constitutes a basis for R^n , so that we can write:

$$x^* = \alpha_0 d_0 + \alpha_1 d_1 + \dots + \alpha_{n-1} d_{n-1}, \quad (12.1)$$

where $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$ are scalar coefficients. By pre-multiplying both members for $d_i^T Q$ we obtain, by the Q -conjugacy assumption and the equation $Qx^* = c$:

$$\alpha_i = \frac{d_i^T Q x^*}{d_i^T Q d_i} = \frac{d_i^T c}{d_i^T Q d_i}. \quad (12.2)$$

From (12.1) and (12.2) we obtain

$$x^* = \sum_{i=0}^{n-1} \frac{d_i^T c}{d_i^T Q d_i} d_i. \quad (12.3)$$

We can note that the coefficients in (12.2) do not require the knowledge of x^* , because of the fact that the directions are mutually Q -conjugate. This would not be true if the set of directions were an arbitrary set of linearly independent directions.

We can consider (12.3) as a finite sequence of steps performed along the directions d_i starting from x_0 with step-size α_i . It can be shown that this corresponds to the minimization of f , in sequence, along the directions d_i for $i = 0, 1, \dots, n-1$.

More specifically, we can establish the following proposition.

Proposition 12.2 (Conjugate Direction Method: Finite Convergence) *Let Q be a symmetric positive definite $n \times n$ matrix and let $\{d_0, d_1, \dots, d_{n-1}\}$ be a system of n non zero vectors mutually conjugate with respect to Q .*

Let f be the quadratic function $f(x) = \frac{1}{2}x^T Qx - c^T x$, and consider the algorithm (conjugate directions method) defined by the iteration

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

where $x_0 \in R^n$ is a given initial point and α_k is the step-size that minimizes f along d_k , given by:

$$\alpha_k = -g_k^T d_k / d_k^T Q d_k = -(Qx_k - c)^T d_k / d_k^T Q d_k.$$

Then:

(i) *if $g_i \neq 0$, for $i = 0, 1, \dots, k-1$ we have:*

$$g_k^T d_i = 0, \quad \text{for } i = 0, 1, \dots, k-1;$$

(ii) *there exists $m \leq n-1$ such that x_{m+1} is the minimum point x^* of f .*

Proof Let $k \geq 1$ and suppose that $g_i \neq 0$, for $i = 0, 1, \dots, k-1$. Starting from a point x_i of the sequence, and using the algorithm defined above, we can write:

$$x_k = x_i + \sum_{j=i}^{k-1} \alpha_j d_j.$$

Pre-multiplying by Q we obtain:

$$Qx_k = Qx_i + \sum_{j=i}^{k-1} \alpha_j Qd_j. \quad (12.4)$$

Recalling that $g(x) = Qx - c$, we can write: $Qx_k - Qx_i = g_k - g_i$. Thus, from (12.4) we get $g_k = g_i + \sum_{j=i}^{k-1} \alpha_j Qd_j$, whence, pre-multiplying by d_i^T , and taking into account the conjugacy assumption, that is assuming $d_i^T Qd_j = 0$ for $i \neq j$, we obtain:

$$\begin{aligned} d_i^T g_k &= d_i^T g_i + \sum_{j=i}^{k-1} \alpha_j d_i^T Qd_j \\ &= d_i^T g_i + \alpha_i d_i^T Qd_i. \end{aligned}$$

Recalling the expression of α_i we obtain $d_i^T g_k = 0$ and hence, by repeating the same reasoning for $i = 0, 1, \dots, k-1$ assertion (i) is proved. Assume now that $g_k \neq 0$, for $k = 0, 1, \dots, n-1$. From (i), for $k = n$ we obtain $g_n^T d_i = 0$, for $i = 0, 1, \dots, n-1$ so that g_n is orthogonal to the n linearly independent vectors d_0, \dots, d_{n-1} . This implies that $g_n = 0$ and that x_n is the global minimizer of f . \square

The preceding proposition shows that if a set of n Q -conjugate directions is available, the global minimizer of the quadratic function can be obtained in at most n steps of the conjugate directions method.

It can be shown that the point x_k generated by the conjugate direction method is the minimizer of f on the affine subspace $x_0 + \mathcal{B}_k$, where \mathcal{B}_k is the linear subspace spanned by the vectors d_0, d_1, \dots, d_{k-1} :

$$\mathcal{B}_k = \left\{ y \in R^n : y = \sum_{i=0}^{k-1} \gamma_i d_i, \quad \gamma_i \in R, \quad i = 0, \dots, k-1 \right\}.$$

The generation of conjugate directions will be considered in the next section.

12.2 The Conjugate Gradient Method (CGM): The Quadratic Case

In Proposition 12.2 we have assumed that the search directions d_0, d_1, \dots, d_{n-1} , mutually conjugate with respect to Q , were already available. Conjugate directions can be constructed by means of a generalization of the *Gram-Schmidt orthogonalization algorithm* and can be given by different expressions. We are interested, in particular, to techniques that do not require storing the matrix Q , but require only the computation of the action of Q on a vector. This will be useful for problems having a favourable sparsity structure.

A method that satisfies this requirement is the *Conjugate Gradient Method* (CGM) of Hestenes and Stiefel, which will be described here.

Let $x_0 \in R^n$ be a given starting point such that $g_0 \neq 0$. We set initially

$$d_0 = -g_0 = -(Qx_0 - c) \quad (12.5)$$

and we define the iteration

$$x_{k+1} = x_k + \alpha_k d_k, \quad (12.6)$$

where α_k is the step-size that minimizes f along d_k , that is

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}. \quad (12.7)$$

If $g_{k+1} \neq 0$, we define the new search direction d_{k+1} by taking:

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k, \quad (12.8)$$

where

$$\beta_{k+1} = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}. \quad (12.9)$$

We can easily verify that this choice guarantees that d_{k+1} is Q -conjugate to d_k ; in fact, from (12.8) to (12.9) we get:

$$d_{k+1}^T Q d_k = 0. \quad (12.10)$$

Before analyzing the properties of this algorithm, we report preliminarily some useful relationships. From the iteration $x_{k+1} = x_k + \alpha_k d_k$, pre-multiplying both members by Q and recalling the expression of the gradient, we obtain:

$$g_{k+1} = g_k + \alpha_k Q d_k. \quad (12.11)$$

As α_k is the optimal step-size, we already know that the directional derivative at x_{k+1} must be zero, so that

$$g_{k+1}^T d_k = 0. \quad (12.12)$$

From (12.8), by replacing $k + 1$ with k , premultiplying by g_k^T and using (12.12) we obtain:

$$g_k^T d_k = -g_k^T g_k. \quad (12.13)$$

From (12.7) and (12.13) it follows immediately that

$$\alpha_k = \frac{g_k^T g_k}{d_k^T Q d_k} = \frac{\|g_k\|^2}{d_k^T Q d_k}. \quad (12.14)$$

We show that the algorithm is well defined, in the sense that $d_k^T Q d_k \neq 0$.

Proposition 12.3 *The CGM computes directions d_k such that $d_k = 0$ if and only if $g_k = 0$. Moreover $\alpha_k = 0$ if and only if $g_k = 0$.*

Proof It follows immediately from (12.13) that $d_k = 0$ implies $g_k = 0$. On the other hand, if $g_k = 0$, by (12.9) we have $\beta_k = 0$ and hence, from (12.8) to (by replacing $k + 1$ with k) we have $d_k = 0$. This proves the first assertion. The second assertion follows from (12.14), which implies $\alpha_k = 0$ if and only if $g_k = 0$. \square

Now we prove that the CGM generates Q -conjugate directions and hence terminates at the minimum point of f .

Proposition 12.4 (CGM: Conjugacy and Convergence) *Let f be the quadratic function defined by $f(x) = \frac{1}{2}x^T Qx - c^T x$, with Q symmetric and positive definite. Suppose $g_0 \neq 0$ and consider the points x_k , for $k = 1, 2, \dots$ generated by the CGM defined by (12.5)–(12.9). Then there exists an integer $m \leq n - 1$ such that, for $i = 1, \dots, m$ we have*

$$g_i^T g_j = 0, \quad j = 0, 1, \dots, i - 1, \quad (12.15)$$

$$d_i^T Q d_j = 0 \quad j = 0, 1, \dots, i - 1, \quad (12.16)$$

$$g_{m+1} = 0. \quad (12.17)$$

Proof As $g_0 \neq 0$ we can assume that there exists an integer $m > 0$ such that $g_i \neq 0$ for $i = 0, 1, \dots, m$ and we show, by induction, that (12.15) and (12.16) hold.

First of all, we show that these relationships hold for $i = 1$. In fact, for $i = 1$, as $d_0 = -g_0$ we have, by (12.12), $g_1^T g_0 = -g_1^T d_0 = 0$ and, by (12.10), $d_1^T Q d_0 = 0$; thus the assertions (12.15) and (12.16) are true. Then we assume that (12.15) and (12.16) hold for a given $i \geq 1$ and we prove that the same is true when we replace i with $i + 1 \leq m$, that is:

$$g_{i+1}^T g_j = 0, \quad j = 0, 1, \dots, i, \quad (12.18)$$

$$d_{i+1}^T Q d_j = 0 \quad j = 0, 1, \dots, i. \quad (12.19)$$

Recalling (12.11) we can write:

$$g_{i+1}^T g_j = g_i^T g_j + \alpha_i g_j^T Q d_i.$$

If $j = 0$ then $g_0 = -d_0$ and hence (12.18) follows from (12.15), (12.16). If $j > 0$ we can write, using (12.8) and (12.11)

$$g_{i+1}^T g_j = g_i^T g_j + \alpha_i (\beta_j d_{j-1} - d_j)^T Q d_i. \quad (12.20)$$

We can distinguish the two cases : $j < i$ and $j = i$. If $j < i$ from (12.15), (12.16) and (12.20) we have immediately $g_{i+1}^T g_j = 0$. If $j = i$ Eq. (12.20) becomes

$$g_{i+1}^T g_i = g_i^T g_i + \alpha_i (\beta_i d_{i-1} - d_i)^T Q d_i.$$

and hence, recalling (12.14) and using again (12.16) we obtain (12.18). Thus we have proved that (12.18) is satisfied, under the assumptions made.

Now we prove that also (12.19) is satisfied, by considering the two cases: $j = i$ and $j < i$. If $j = i$, the definition of β_{i+1} guarantees that d_{i+1} is conjugate to d_i , so that (12.19) is true. If $j < i$, from (12.8) we get

$$d_{i+1} = -g_{i+1} + \beta_{i+1} d_i,$$

and we can write:

$$d_{i+1}^T Q d_j = -g_{i+1}^T Q d_j + \beta_{i+1} d_i^T Q d_j,$$

whence it follows, taking into account (12.11), rewritten with $k = j$,

$$d_{i+1}^T Q d_j = -g_{i+1}^T (g_{j+1} - g_j) \frac{1}{\alpha_j} + \beta_{i+1} d_i^T Q d_j. \quad (12.21)$$

As $j < i$ we have also $j + 1 < i + 1$ and hence, by (12.18) (which we have already proved) and the assumption (12.16), we get from (12.21) $d_{i+1}^T Q d_j = 0$. This completes the induction and therefore (12.15), (12.16) must hold for every $i \leq m \leq n - 1$. It follows that the directions generated by the algorithm are mutually Q -conjugate and thus, by Proposition 12.2, there must exist $m \leq n - 1$ such that $g(x_{m+1}) = 0$, so that the point x_{m+1} is the minimizer of f . \square

We observe that the expression of β_{k+1} can be simplified in a way that the Hessian matrix does not appear explicitly and this will be useful in the extension of the CGM to non quadratic problems.

From (12.11) we can obtain

$$Q d_k = (g_{k+1} - g_k) / \alpha_k$$

and hence we can rewrite (12.9) as

$$\beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k) / \alpha_k}{d_k^T (g_{k+1} - g_k) / \alpha_k} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}. \quad (12.22)$$

From (12.22), taking into account (12.12), we have:

$$\beta_{k+1} = -\frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T g_k}. \quad (12.23)$$

Using (12.13), from (12.23) we get

$$\beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{\|g_k\|^2}. \quad (12.24)$$

From (12.24), as $g_{k+1}^T g_k = 0$, by (12.15), we have also:

$$\beta_{k+1} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2}. \quad (12.25)$$

Then the CGM can be described with the following conceptual model.

Algorithm 12.1 (Conjugate Gradient Method (CGM))

Data: $x_0 \in R^n$.

Compute $g_0 = Qx_0 - c$, $d_0 = -g_0$ and set $k = 0$.

While $g_k \neq 0$ compute:

$$\alpha_k = \|g_k\|^2 / d_k^T Q d_k,$$

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

$$g_{k+1} = g_k + \alpha_k Q d_k,$$

$$\beta_{k+1} = \|g_{k+1}\|^2 / \|g_k\|^2,$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k.$$

Set $k = k + 1$

End While

Remark 12.1 Consider a quadratic function

$$f(x) = \frac{1}{2} x^T Q x - c^T x \quad (12.26)$$

where Q is symmetric and positive semidefinite. Assume that f admits minimum, i.e., the linear system $Qx = c$ admits solution. It is possible to show (see the Appendix) that CGM converges to a minimum of f in at most n iterations. Then, CGM can be used to compute a solution of linear least squares problems, i.e., problems whose objective function is

$$f(x) = \frac{1}{2} \|Ax - b\|^2, \quad (12.27)$$

being A any matrix $m \times n$ and $b \in R^m$. Indeed, up to an additive constant, (12.27) is a quadratic function of the form (12.26) with $Q = A^T A$, $c = A^T b$ and, as shown in Chap. 2, we have that f admits minimum, i.e., the linear system

$$A^T Ax = A^T b$$

admits a solution. \square

We report below the scheme of the conjugate gradient method for linear least squares problems.

Algorithm 12.2 Conjugate Gradient Method for Linear Least Squares

Data: $x_0 \in R^n$, $tol \geq 0$

Set $r_0 = b - Ax_0$, $d_0 = -g_0 = A^T r_0$, $\gamma_0 = \|g_0\|^2$, $k = 0$.

While $\gamma_k \geq tol$

Set

$$q_k = Ad_k, \quad (12.28)$$

$$\alpha_k = \gamma_k / \|q_k\|^2, \quad (12.29)$$

$$r_{k+1} = r_k - \alpha_k q_k, \quad (12.30)$$

$$-g_{k+1} = A^T r_{k+1}, \quad (12.31)$$

$$\gamma_{k+1} = \|g_{k+1}\|^2, \quad (12.32)$$

$$\beta_k = \gamma_{k+1} / \gamma_k, \quad (12.33)$$

(continued)

Algorithm 12.2 (continued)

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad (12.34)$$

$$k = k + 1.$$

End While

12.3 Convergence Rate

It has been shown in the preceding section that the CGM terminates in n iterations at most, but this could be not very useful for large values of n . In fact, the CGM method, viewed as a *direct method*, can be convenient only for sparse matrices, when the matrix-vector products requested in the method can be efficiently evaluated. In the general case, the number of algebraic operations required for computing a solution is of the order of n^3 and the Cholesky factorization is, in general, less expensive. However, one of the most important features of the CGM is the fact that it can be viewed also as an *iterative method* such that the *residual* $g = Qx - c$ of the linear equation $Qx - c = 0$, converges to zero. Thus several studies have been carried out on the convergence rate of the CGM and we give here a short account of some of the main results. One of the first estimates, apparently due to Daniel [57] is

$$\|x_k - x^*\| \leq 2 \left(\frac{\lambda_M}{\lambda_m} \right)^{1/2} \left(\frac{\sqrt{\lambda_M} - \sqrt{\lambda_m}}{\sqrt{\lambda_M} + \sqrt{\lambda_m}} \right)^k \|x_0 - x^*\|, \quad (12.35)$$

where λ_M and λ_m are, respectively, the maximum and the minimum eigenvalue of Q . By comparing this expression with the estimate given for the steepest descent method in the quadratic case, it can be easily seen that now the convergence rate depends on the *root* of the condition number and this indicates that, in general, the CGM may have a better convergence speed than the gradient method.

Another important result is related to the dependency of the convergence speed on the structure of the spectrum of the matrix Q . To illustrate this point, let us denote by $\|x\|_Q$ the norm $\|x\|_Q = (x^T Q x)^{1/2}$, so that we can write

$$\frac{1}{2} \|x - x^*\|_Q^2 = \frac{1}{2} (x - x^*)^T Q (x - x^*) = f(x) - f(x^*). \quad (12.36)$$

Then we can state the following proposition, whose proof can be found, for instance, in [16].

Proposition 12.5 Let $\Omega = \{\lambda_1, \dots, \lambda_p\}$, with $p \leq n$, the set of distinct eigenvalues of Q . Suppose that $p - k$ elements of Ω are contained in the interval $[a, b]$ and that the remaining k distinct eigenvalues are greater than b . Then, if x_{k+1} is the point generated by the CGM after k iterations, we have.

$$\|x_{k+1} - x^*\|_Q^2 \leq \left(\frac{b-a}{b+a}\right)^2 \|x_0 - x^*\|_Q^2. \quad (12.37)$$

□

The preceding proposition shows that the first k iterations essentially eliminate the effect on the error of the k largest eigenvalues, so that, if $b - a$ is small, then a good reduction of the error is obtained in k iterations. We note that the condition number may be not much relevant when most of the eigenvalues are clustered in a small interval.

As immediate consequence of Proposition 12.5 we get the following result.

Corollary 12.1 Suppose that the number of distinct eigenvalues of Q is k . Then the conjugate gradient method converges in at most k iterations.

Proof The proof follows from (12.37) setting a equal to any value smaller than the minimum eigenvalue of Q and setting $b = a$. □

12.4 Preconditioning

The results given in the preceding section show that the efficiency of the CGM depends, essentially, on the structure of the spectrum of Q . Preconditioning techniques have the objective of transforming the problem into an equivalent problem with a matrix with eigenvalues clustered as much as possible near one.

Ideally, if M is a symmetric positive definite matrix that approximates Q^{-1} , then the system

$$MQx = Mc$$

would be equivalent to the original system and the matrix MQ would have all the eigenvalues approximately equal to one. However, there is no guarantee that MQ is a symmetric positive definite matrix. We can avoid this problem by expressing the

matrix Q through its square roots, that is letting $Q = B^2$, so that the system can be written in the form

$$BBx = c,$$

and hence, if $S \approx B^{-1}$ and we set $y = S^{-1}x$, then, pre-multiplying both members by S we obtain the equivalent system

$$SQSy = Sc, \quad (12.38)$$

where now SQS is a symmetric positive matrix that approximates the identity. Thus, if the CGM method is used for computing the solution y^* of (12.38), we get the solution of the original system by letting $x^* = Sy^*$.

Actually, computations can be organized without forming and computing the matrix SQS and by updating directly the current solution x_k of the original system. Let us write the formulae of the CGM method applied to (12.38). For $k \geq 0$, denoting by \tilde{d}_k and \tilde{g}_k the search direction and the residual in the CGM for system (12.38), we can write

$$y_{k+1} = y_k + \alpha_k \tilde{d}_k, \quad (12.39)$$

where

$$\alpha_k = \frac{\tilde{g}_k^T \tilde{g}_k}{\tilde{d}_k^T S Q S \tilde{d}_k}, \quad (12.40)$$

$$\tilde{g}_k = SQSy_k - Sc = S(QSy_k - c) \quad (12.41)$$

Pre-multiplying both members of (12.39) by S and using the transformation $x = Sy$, we have

$$x_{k+1} = x_k + \alpha_k S \tilde{d}_k,$$

whence, letting

$$d_k = S \tilde{d}_k, \quad (12.42)$$

we obtain

$$x_{k+1} = x_k + \alpha_k d_k.$$

Equation (12.41) can be rewritten as

$$\tilde{g}_k = S(Qx_k - c) = Sg_k, \quad (12.43)$$

and then, from (12.40), using (12.42) and letting $M = S^2$, we can write:

$$\alpha_k = \frac{g_k^T S S g_k}{d_k^T Q d_k} = \frac{g_k^T M g_k}{d_k^T Q d_k}. \quad (12.44)$$

We have also

$$\tilde{d}_0 = -\tilde{g}_0, \quad (12.45)$$

$$\tilde{d}_k = -\tilde{g}_k + \beta_k \tilde{d}_{k-1} \quad k \geq 1, \quad (12.46)$$

where

$$\beta_k = \frac{\|\tilde{g}_k\|^2}{\|\tilde{g}_{k-1}\|^2} = \frac{g_k^T S S g_k}{g_{k-1}^T S S g_{k-1}} = \frac{g_k^T M g_k}{g_{k-1}^T M g_{k-1}}. \quad (12.47)$$

From (12.45) and (12.46), multiplying by S , we get

$$d_0 = -S\tilde{g}_0 = -SSg_0 = -Mg_0 \quad (12.48)$$

$$d_k = -g_k + \beta_k d_{k-1} \quad k \geq 1. \quad (12.49)$$

Now, by defining the vector $z_k = Mg_k$ and the scalar $\tau_k = z_k^T g_k$, from (12.44) and (12.47) we obtain the expressions

$$\alpha_k = \frac{\tau_k}{d_k^T Q d_k}$$

$$\beta_{k+1} = \frac{\tau_{k+1}}{\tau_k},$$

which are employed in the following scheme.

Algorithm 12.3 (Preconditioned CGM)

Data: starting point $x_0 \in R^n$, M symmetric and positive definite $n \times n$ matrix.

Set $g_0 = Qx_0 - c$, $d_0 = -Mg_0$, $k = 0$.

While $g_k \neq 0$

 Set

$$z_k = Mg_k \quad \tau_k = z_k^T g_k, \quad (12.50)$$

(continued)

Algorithm 12.3 (continued)

$$\alpha_k = \frac{\tau_k}{d_k^T Q d_k}, \quad (12.51)$$

$$x_{k+1} = x_k + \alpha_k d_k, \quad (12.52)$$

$$g_{k+1} = g_k + \alpha_k Q d_k, \quad (12.53)$$

$$z_{k+1} = M g_{k+1} \quad \tau_{k+1} = z_{k+1}^T g_{k+1}, \quad (12.54)$$

$$\beta_{k+1} = \frac{\tau_{k+1}}{\tau_k}, \quad (12.55)$$

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k, \quad (12.56)$$

$$k = k + 1.$$

End While

Various different choices have been employed for the choice of preconditioners, which are often designed in relation to the class of linear systems to be solved. An easily implemented preconditioner for general classes of matrices is the so called *Jacobi preconditioner*, consisting in the inverse of the diagonal of Q . However, various more efficient choices have been proposed and the reader is addressed to the references cited at the end of the chapter.

12.5 The CGM in the Non Quadratic Case

The extension of the CGM in the non quadratic case consists in modifying the CGM algorithm introduced before, in a way that it can be defined without using the knowledge of the Hessian matrix, but the algorithm is exactly the CGM when f is quadratic. This requires that:

- the search direction is defined by adopting a formula for β_{k+1} , which is valid in the quadratic case, but does not contain the matrix Q ;
- the analytical expression of α_k is replaced by a line search algorithm.

The various algorithms proposed in the non quadratic case differ essentially for the formula adopted for β_{k+1} and for the linesearch employed.

We have already seen in the quadratic case that formula (12.9) can put into the form

$$\beta_{k+1} = \frac{g_{k+1}^T y_k}{d_k^T y_k}, \quad (12.57)$$

where we set, to simplify notation: $y_k = g_{k+1} - g_k$. Moreover, in the quadratic case, we have already seen that we can give the following equivalent expressions for numerator and denominator in (12.57):

$$g_{k+1}^T y_k = \|g_{k+1}\|^2 \quad (12.58)$$

$$d_k^T y_k = -d_k^T g_k = \|g_k\|^2. \quad (12.59)$$

By combining into the six possible ways the expressions above we obtain the following formulae, where we also indicate the authors that have proposed or studied each formula.

$\beta_{k+1}^{\text{HS}} = \frac{g_{k+1}^T y_k}{d_k^T y_k}$	(Hestenes-Stiefel)
$\beta_{k+1}^{\text{FR}} = \frac{\ g_{k+1}\ ^2}{\ g_k\ ^2}$	(Fletcher-Reeves)
$\beta_{k+1}^{\text{PPR}} = \frac{g_{k+1}^T y_k}{\ g_k\ ^2}$	(Polyak-Polak-Ribiére),
$\beta_{k+1}^{\text{F}} = \frac{\ g_{k+1}\ ^2}{-d_k^T g_k}$	(Fletcher)
$\beta_{k+1}^{\text{LS}} = \frac{g_{k+1}^T y_k}{-d_k^T g_k}$	(Liu-Storey)
$\beta_{k+1}^{\text{DY}} = \frac{\ g_{k+1}\ ^2}{d_k^T y_k}$	(Day-Yuan)

The best known formulae are that of Fletcher and Reeves (FR) and that of Polyak-Polak-Ribiére (PPR), which is the preferred formula in most of codes. We can note that many other equivalent formulae can be defined by adding terms that vanish in the quadratic case. Thus, if $\tilde{\beta}_{k+1}$ is any formula equivalent to (12.57), then every formula of the type:

$$\beta_{k+1} = \tilde{\beta}_{k+1} + \xi_k g_{k+1}^T d_k + \zeta_k g_{k+1}^T g_k, \quad \xi_k, \zeta_k \in R,$$

where ξ_k and ζ_k are arbitrary scalars, will be still equivalent to (12.57) in the quadratic case. As an example, the formula proposed by Hager and Zhang [137] can be put into the form

$$\begin{aligned}\beta_{k+1}^{\text{HZ}} &= \left(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T \frac{g_{k+1}}{d_k^T y_k} \\ &= \beta_{k+1}^{\text{HS}} - 2 \frac{\|y_k\|^2}{(d_k^T y_k)^2} g_{k+1}^T d_k.\end{aligned}$$

Moreover, as in the quadratic case we have $\beta_{k+1} \geq 0$, if β_{k+1} is any formula equivalent to (12.57) also the formula

$$\beta_{k+1}^+ = \max\{0, \beta_{k+1}\}$$

will be equivalent.

In the non quadratic case, the different formulae correspond to different algorithms and the choice of the step-size α_k must be related to the specific formula adopted for β_{k+1} . In fact, in order to define a descent technique a first requirement is obviously that the search direction

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$$

is a descent direction. Then, a sufficient descent condition is that

$$-g_{k+1}^T g_{k+1} + \beta_{k+1} g_{k+1}^T d_k < 0.$$

As β_{k+1} depends on $g_{k+1} = g(x_k + \alpha_k d_k)$, the line search performed along d_k for computing α_k must take into account the descent condition on d_{k+1} . We note that this condition can be satisfied either for a sufficiently small value of β_{k+1} or through an appropriate line search (or a sequence of line searches) that yields a sufficiently small value of $g_{k+1}^T d_k$.

However, as we already know, satisfaction of the descent conditions is not enough for guaranteeing global convergence and appropriate further conditions must be imposed on the line search or, alternatively, the search direction must be modified when required.

Some of the simplest ideas for enforcing global convergence can be that of *restarting* the CGM periodically along the negative gradient direction or that of resorting to the negative gradient direction when some sufficient convergence condition is violated. We know that if the gradient direction is used for an infinite subsequence then we can guarantee, under usual assumptions, the existence of a limit point that is a stationary point, provided that we have $f(x_{k+1}) \leq f(x_k)$ for all k . We can easily define restarting rules such that the method is not modified in the quadratic case. However, computational experience has indicated that frequent

restarts are not convenient and may destroy the efficiency of the CGM. Thus, several attempts have been made to guarantee global convergence without (explicit) restarts. Some results on the global convergence of the FR method and of the PPR method are shortly discussed in the next sections.

12.6 Fletcher-Reeves Method

In this section we assume that $\beta_{k+1} = \beta_{k+1}^{\text{FR}} = \|g_{k+1}\|^2/\|g_k\|^2$ and we will define as Fletcher-Reeves (FR) method a conjugate gradient method where β_{k+1} is defined through this formula. The global convergence of the FR method has been established both when the line search is exact and when we use an inexact line search based on Wolfe conditions. In the case of exact line search we have the following result, which we state without proof. See, for instance, [96].

Proposition 12.6 (Convergence of the FR Method with Exact Line Search) *Let $f : R^n \rightarrow R$ be continuously differentiable with Lipschitz continuous gradient on an open convex set \mathcal{D} containing the level set \mathcal{L}_0 , and suppose that \mathcal{L}_0 is compact. Let $\{x_k\}$ be an infinite sequence with $g_k \neq 0$, generated by the Fletcher-Reeves method, where $\alpha_k = \text{Arg min}_{\alpha \geq 0} f(x_k + \alpha d_k)$. Then there exists a limit point which is a stationary point of f . \square*

The extension of this result to the case of inexact line searches is due to Al-Baali [1] and is based on the adoption of the strong Wolfe conditions. More specifically, we can consider the following scheme.

Algorithm 12.4 (Fletcher-Reeves Method with Strong Wolfe Line Search)

Data: $x_0 \in R^n$, $0 < \gamma < \sigma < 1/2$.

Compute g_0 and set $d_0 = -g_0$, $k = 0$.

While $g_k \neq 0$

 Compute α_k such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k g_k^T d_k, \quad (12.60)$$

$$|g_{k+1}^T d_k| \leq \sigma |g_k^T d_k|. \quad (12.61)$$

(continued)

Algorithm 12.4 (continued)

Set $x_{k+1} = x_k + \alpha_k d_k$.
 Compute $\beta_{k+1} = \|g_{k+1}\|^2 / \|g_k\|^2$
 Set $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$.
 Set $k = k + 1$.

End While

The convergence of this algorithm is stated in the next proposition [1].

Proposition 12.7 (Convergence of the FR Method with Wolfe line search)

Let $f : R^n \rightarrow R$ be continuously differentiable with Lipschitz continuous gradient on an open convex set \mathcal{D} containing the level set \mathcal{L}_0 , and suppose that \mathcal{L}_0 is compact. Let $\{x_k\}$ be an infinite sequence with $g_k \neq 0$, generated by the Fletcher-Reeves method, where the step-size α_k satisfies conditions (12.60), (12.61) with $\sigma \in (0, 1/2)$. Then there exists a limit point of $\{x_k\}$ which is a stationary point of f . \square

Computational experience has shown that the FR method is not, in general, among the most efficient conjugate gradient techniques, as the algorithm can get stuck in difficult regions,

12.7 Method of Polyak-Polak-Ribière (PPR)

The PPR method is the technique commonly preferred among the conjugate gradient methods. In this section we suppose, unless otherwise stated, that

$$\beta_{k+1} = \beta_{k+1}^{\text{PPR}} = \frac{g_{k+1}^T y_k}{\|g_k\|^2}, \quad \text{where } y_k = g_{k+1} - g_k. \quad (12.62)$$

The structure of β_{k+1} suggests some explanation of the greater efficiency of the PRP method in comparison with the FR method. It was observed by Powell that if x_{k+1} is not much different from x_k and we have $g_{k+1} \approx g_k$ then $\beta_{k+1} \approx 0$ and hence $d_{k+1} \approx -g_{k+1}$. Thus the PRP method possesses a sort of automatic restart that may avoid some of the difficulties encountered by the FR method in difficult cases.

The global convergence of the PRP method with exact line searches has been first established in the convex case. More specifically the following result holds [207].

Proposition 12.8 (Convergence PPR Method: Convex Case) Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable on an open convex set \mathcal{D} containing the level set \mathcal{L}_0 , and assume there exist numbers $0 < m \leq M$ such that

$$m\|h\|^2 \leq h^T \nabla^2 f(x)h \leq M\|h\|^2, \quad \text{for all } x \in \mathcal{L}_0, h \in \mathbb{R}^n. \quad (12.63)$$

Suppose that \mathcal{L}_0 is compact. Let $\{x_k\}$ be an infinite sequence, with $g_k \neq 0$, generated by the PPR method, where the step-size α_k is defined by

$$\alpha_k = \operatorname{Arg} \min_{\alpha \geq 0} f(x_k + \alpha d_k).$$

Then the sequence $\{x_k\}$ converges to the minimum point of f on \mathbb{R}^n . □

In the non convex case, establishing the global convergence of the PPR method is more difficult than in the case of the FR method. In fact, some counterexamples have been found [217], where an exact line search may lead to a failure of the method.

Moreover, even guaranteeing that d_{k+1} is a descent direction is not possible, in the general case, using standard line search techniques, such as, for instance, those based on Wolfe conditions. However, when required, we can modify Wolfe's conditions in order to determine a step-size that satisfies a descent condition.

This is illustrated in the following algorithm model, where we assume that $x_k \in \mathcal{L}_0$, that \mathcal{L}_0 is compact, and that d_k is a given search direction such that $g_k^T d_k < 0$.

Algorithm WM (Wolfe Modified)

Data $1/2 > \gamma > 0, \sigma > \gamma, \varepsilon_k > 0, \delta_1 \in (0, 1)$.

1. Compute η_k such that:

- (i) $f(x_k + \eta_k d_k) \leq f_k + \gamma \eta_k g_k^T d_k,$
- (ii) $g(x_k + \eta_k d_k)^T d_k \geq \sigma g_k^T d_k \quad (\text{or: } |g(x_k + \eta_k d_k)^T d_k| \leq \sigma |g_k^T d_k|)$

2. Compute α_k such that either $\|g(x_k + \alpha_k d_k)\| \leq \varepsilon_k$ or we have that the vectors

$x_{k+1} = x_k + \alpha_k d_k$ and $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$ satisfy:

- (a₁) $f_{k+1} \leq f(x_k + \eta_k d_k),$
- (a₂) $g_{k+1}^T d_{k+1} \leq -\delta_1 \|g_{k+1}\|^2 < 0.$

It can be easily verified that Algorithm WM is well defined under the assumptions stated. In fact, we already know that there exists a finite procedure for computing a

scalar η_k where the Wolfe conditions are satisfied. Starting from this point we can define a convergent minimization process (for instance the steepest descent method using again Wolfe conditions referred to α) that generates a sequence of step-sizes $\alpha(j)$, for $j = 0, 1, \dots$ with $\alpha(0) = \eta_k$ and such that for $j \rightarrow \infty$ we have

$$f(x_k + \alpha(j)d_k) < f(x_k + \eta_k d_k) \quad \text{and} \quad g(x_k + \alpha(j)d_k)^T d_k \rightarrow 0.$$

Recalling the PPR formula and taking into account the compactness of \mathcal{L}_0 , we can easily establish that conditions (a₁) and (a₂) are satisfied in a finite numbers of j -iterations, unless $\|g(x_k + \alpha(j)d_k)\|$ converges to 0, but in this case we have necessarily $\|g(x_k + \alpha(j)d_k)\| \leq \varepsilon_k$ for sufficiently large j .

As regards global convergence, in order to establish convergence results, while retaining as much as possible the good features of the method, we can follow two different approaches:

- the formula for β_{k+1} is modified and a suitable line search algorithm is defined, which can also be an exact line search algorithm;
- the formula for β_{k+1} is the PPR formula, but an appropriate line search is defined (which must be necessarily inexact in certain cases).

A modified PPR method with global convergence properties has been defined [111] by letting $\beta_k = \max\{\beta_k^{\text{PPR}}, 0\}$, as suggested by Powell [210], and employing an exact linesearch or an implementable inexact linesearch, essentially equivalent to Algorithm WM defined above, that guarantees a descent condition of the form given in (a₂) of Algorithm WM.

An alternative approach, which does not require restarting along the steepest descent direction and hence leaves unmodified the PPR search direction, can be based on the adoption of line search rules that, in addition to some standard acceptance condition, ensure the limit $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$, and guarantee that d_{k+1} satisfies a descent condition. The simplest techniques based on this approach can be obtained through simple modifications of Armijo's method. A first possibility is that of imposing an upper bound on the initial step-size Δ_k and a descent condition on d_{k+1} , as in the following scheme.

Armijo Modified (AM) Algorithm

Data. $\rho_2 > \rho_1 > 0$, $1 > \gamma > 0$, $\delta \in (0, 1)$, $\theta \in (0, 1)$.

1. Set $\tau_k = \frac{|g_k^T d_k|}{\|d_k\|^2}$ and choose $\Delta_k \in [\rho_1 \tau_k, \rho_2 \tau_k]$.
2. Determine $\alpha_k = \max\{\theta^j \Delta_k, j = 0, 1, \dots\}$ such that the vectors

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k d_k, \\ d_{k+1} &= -g_{k+1} + \beta_{k+1} d_k, \end{aligned}$$

(continued)

satisfy:

- (i) $f_{k+1} \leq f_k + \gamma \alpha_k g_k^T d_k;$
- (ii) $g_{k+1}^T d_{k+1} < -\delta \|g_{k+1}\|^2 < 0$

A different possibility can be that of relaxing the bound on the step-size, by requiring $\Delta_k \geq \rho |g_k^T d_k| / \|d_k\|^2$, and replacing condition (i) at step 2 with a *parabolic acceptance rule* of the form

$$f_{k+1} \leq f_k - \gamma \alpha_k^2 \|d_k\|^2.$$

It can be shown [127] that these algorithms are well defined and guarantee the existence of a limit point which is a stationary point, under the assumptions that \mathcal{L}_0 is compact and that the gradient is Lipschitz continuous. A potential limitation is that the acceptability conditions could exclude, in principle, that accurate line searches can be performed and this, although in some sense necessary on the basis of the counterexamples, could deteriorate the behaviour of the PPR method in non pathological problems. In practice, we could relax the acceptability conditions by employing some adaptive choice of the parameters. In particular, if we set

$$\psi_k = \min\{1, \|g_k\|\}^\tau,$$

for some $\tau > 0$, we can replace the parameters $\rho_1, \rho_2, \rho, \gamma$ introduced above with $\rho_1 \psi_k, \rho_2 \psi_k, \rho \psi_k, \gamma \psi_k$ in a way that the conditions become less restrictive as the gradient converges to zero.

However, in the quadratic case we have no guarantee that the algorithm can be identified, right at the start, with the PPR algorithm.

A different approach can be defined by choosing the appropriate line search algorithm on the basis of the norm of the search direction. In the scheme reported below the step-size can be computed using Wolfe conditions or also an exact line search whenever $\|d_k\| \leq b_k$, where b_k is a bound valid in the quadratic case; when $\|d_k\| > b_k$ an Armijo-type algorithm is employed. We will refer to the modified Wolfe's condition defined before with the notation WM(ε), to indicate that the step-size is computed with termination criterion at step 2 defined by ε . To simplify our description, we assume here that the test $\|g_k\| \leq \varepsilon$ is a termination test for the algorithm.

Algorithm PPR

Data. starting point $x_0 \in R^n$, tolerance $\varepsilon > 0$.

Set $d_0 = -g_0$ and $k = 0$.

While $\|g_k\| \neq 0$:

If $\|d_k\| \leq b_k$ then:

compute α_k and β_{k+1} by means of Algorithm WM(ε);

if test $\|g(x_k + \alpha_k d_k)\| \leq \varepsilon$ is satisfied then

terminate at $x_k + \alpha_k d_k$ and exit

else set

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

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k,$$

end if

Else, (if $\|d_k\| > b_k$), compute α_k using algorithm AM and set

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

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k.$$

End if

Set $k = k + 1$

End While

As regards the definition of the bound b_k a simple rule can be that of using the bound that holds for the CGM in the quadratic case [1], that is:

$$\|d_k\|^2 = \|g_k\|^4 \sum_{j=0}^k \|g_{k-j}\|^{-2},$$

and hence we can assume, for instance:

$$b_k = b \|g_k\|^2 \left(\sum_{j=0}^{\min\{k,n\}} \|g_{k-j}\|^{-2} \right)^{1/2}, \quad (12.64)$$

where $b \geq 1$ is a given constant.

The convergence of the algorithm is stated in the following proposition, whose proof is given in [128].

Proposition 12.9 (Convergence of Algorithm PPR) *Let $f : R^n \rightarrow R$ be continuously differentiable with Lipschitz continuous gradient on an open convex set \mathcal{D} containing the level set \mathcal{L}_0 , and suppose that \mathcal{L}_0 is compact. Let $\{x_k\}$ be an infinite sequence, generated by Algorithm PPR where b_k is defined by (12.64). Then either the algorithm terminates in a finite number of iterations or there exists a limit point of $\{x_k\}$ which is a stationary point of f .* \square

Algorithm PPR permits exact line searches, as long as $\|d_k\|$ does not exceed the bound b_k . Thus the algorithm reduces to the CGM of Hestenes-Stiefel in the quadratic case, provided that we compute the optimal step-size at the start of line searches. This can be easily done without requiring the knowledge of the Hessian matrix, by performing a new function evaluation along d_k and then employing a (safeguarded) quadratic interpolation formula. As the bound b_k is valid in the quadratic case, the algorithm will coincide with the linear CGM during the first n iterations. When n is small it could be convenient to operate a restart.

In any case, it would seem that the algorithm defined above is the only algorithm such that:

- the PPR formula is not modified;
- convergence in the non quadratic case is guaranteed under usual assumptions;
- the linear CGM method is reobtained in the quadratic case.

Other approaches can be based on various modifications of the PPR formula [128].

12.8 Appendix: The CG Method When the Hessian Matrix is Positive Semidefinite

Consider the problem

$$\min_{x \in R^n} f(x) = \frac{1}{2} x^T Q x - c^T x, \quad (12.65)$$

where Q is a symmetric positive semidefinite matrix, and assume that it admits solution. We prove that the conjugate gradient method converges to a solution of (12.65).

To this aim we recall that the *null space* of Q is the following linear subspace

$$\mathcal{N}(Q) = \{x \in R^n : Qx = 0\},$$

and that the *range space* of Q is the linear subspace

$$\mathcal{R}(Q) = \{x \in R^n : x = Qy, y \in R^n\}.$$

From known linear algebra results we have

$$\mathcal{R}(Q) \cap \mathcal{N}(Q) = 0. \quad (12.66)$$

The convergence analysis differs from that presented in the standard case, where the matrix Q is assumed to be positive definite, since now it may happen that $d_k^T Q d_k = 0$, being Q semidefinite positive.

Proposition 12.10 *Let Q be a symmetric positive semidefinite matrix and assume that problem (12.65) admits a solution. Then the conjugate gradient method converges in at most n iterations to a solution of (12.65).*

Proof In order to prove the thesis it is sufficient to show that

$$d_k^T Q d_k = 0 \quad \text{implies} \quad g_k = 0. \quad (12.67)$$

Indeed, if $d_k^T Q d_k > 0$ for every $k \geq 0$, then we can repeat the reasonings used in the proof of Proposition 12.4 by obtaining the same conclusions.

First, by induction, we show that for every $k \geq 0$ we have

$$g_k, d_k \in \mathcal{R}(Q), \quad (12.68)$$

where $\mathcal{R}(Q)$ is the range space of Q .

Condition (12.68) is true for $k = 0$. Indeed, problem (12.65) admits solution, therefore there exists a point x^* such that

$$Qx^* = c.$$

Then $c \in \mathcal{R}(Q)$ and hence it follows that $g_0 = Qx_0 - c \in \mathcal{R}(Q)$. Moreover, as $d_0 = -g_0$, we have $d_0 \in \mathcal{R}(Q)$.

Assume that (12.68) holds for $k - 1$ (with $k \geq 1$): we prove by induction that it holds for k . We have

$$g_k = g_{k-1} + \alpha_{k-1} Q d_{k-1},$$

from which it follows that $g_k \in \mathcal{R}(Q)$. Then, as

$$d_k = -g_k + \beta_k d_{k-1},$$

we have that $d_k \in \mathcal{R}(Q)$.

To prove (12.67) suppose $d_k^T Q d_k = 0$. We can prove that d_k belongs to the null space of Q , i.e., $d_k \in \mathcal{N}(Q)$. Indeed, by using the spectral decomposition of Q , we can write

$$d_k^T Q d_k = \sum_{i=1}^n \lambda_i d_k^T u_i u_i^T d_k = \sum_{i=1}^n \lambda_i (u_i^T d_k)^2 = 0,$$

being λ_i, u_i , for $i = 1, \dots, n$, the eigenvalues and the eigenvectors of Q , respectively. Since $\lambda_i \geq 0$, we must have $\lambda_i (u_i^T d_k) = 0$ for $i = 1, \dots, n$. Then we have

$$Q d_k = \sum_{i=1}^n u_i \lambda_i (u_i^T d_k) = 0,$$

i.e., $d_k \in \mathcal{N}(Q)$.

Using (12.68) it follows that

$$d_k \in \mathcal{R}(Q) \cap \mathcal{N}(Q),$$

and hence, from (12.66), we obtain $d_k = 0$.

Recalling that $g_k^T d_k = -\|g_k\|^2$ (see (12.13)) we have that $g_k = 0$ and hence we can conclude that x_k is a solution of (12.65). \square

12.9 Exercises

12.1 Define a computer code based on the conjugate gradient method for the minimization of strictly convex quadratic functions. In particular, consider problems with diagonal Hessian matrix and study the behavior of the method in dependence of the condition number of the matrix

12.2 Define a computer code based on the PPR method and study the behavior of the method on some non quadratic test problem.

12.3 Define a computer code for the solution of linear least squares problems $\min_{1/2} \|Ax - b\|^2$.

12.10 Notes and References

Conjugate direction methods were originally introduced as methods for solving linear systems with symmetric positive definite matrix [143]. A relevant feature is that these methods actually can be implemented as iterative methods, by specifying

only the product Qu for any given u and this can be useful for sparse or structured systems. The literature is very large and we can only mention some basic references. For the linear conjugate gradient method we refer, in particular, to [119, 142, 152, 177] and to the references quoted in these books. In the non quadratic case the FR method was originally introduced in [96]. The PPR method was proposed independently in [209] and in [207]. References to subsequent developments can be found, for instance, in [137] and in [222]. Global convergence of the FR method with inexact line searches has been established in [1]. Convergence conditions for modified PPR algorithms have been given in [111]. Global convergence for the PPR method based on modified linesearch techniques have been given in [127] and [128]. Useful references on preconditioning techniques are [25, 119, 222, 232].

Chapter 13

Newton's Method



We report here some basic results on Newton-type methods. First we analyze local convergence properties of these methods in the solution of a system of nonlinear equations and we extend the results to unconstrained minimization problems. Then we consider some globally convergent algorithms based on modifications of the Newton search direction and in the adoption of line searches. Non monotone, linesearch based, globalization techniques will be considered in Chap. 24.

13.1 The Pure Newton Iteration

Newton's method has been originally studied in relation to the solution of a system of nonlinear equations in function spaces [149]. Here we consider the system

$$F(x) = 0,$$

where $F : R^n \rightarrow R^n$ is a continuously differentiable vector function with components $F_i : R^n \rightarrow R$, on the basis of [200].

We denote by $J(x)$ the Jacobian matrix of F evaluated at x , that is the $n \times n$ matrix with elements

$$J_{ij}(x) = \frac{\partial F_i(x)}{\partial x_j}, \quad i, j = 1, \dots, n.$$

Starting from an initial point $x_0 \in R^n$, Newton's method generates a sequence $\{x_k\}$, by solving at each step a linear system that approximates the given system.

More specifically, as F is continuously differentiable, given $x_k \in R^n$, we can write, for every $s \in R^n$:

$$F(x_k + s) = F(x_k) + J(x_k)s + \gamma(x_k, s)$$

where $\gamma(x_k, s)/\|s\| \rightarrow 0$ for $s \rightarrow 0$. Therefore, for “small” values of $\|s\|$, we can attempt to determine a vector s_k such that $F(x_k + s_k) \cong 0$, by solving the linear system

$$F(x_k) + J(x_k)s = 0.$$

If $J(x_k)$ is nonsingular, the solution of the linear system is

$$s_k = -[J(x_k)]^{-1}F(x_k)$$

and hence the (pure) Newton's iteration becomes

$$x_{k+1} = x_k - [J(x_k)]^{-1}F(x_k). \quad (13.1)$$

In an unconstrained optimization problem Newton's method can be viewed as a method for solving the system of n equations

$$\nabla f(x) = 0,$$

that yields stationary points of f . When f is convex this is equivalent to construct a sequence $\{x_k\}$, by minimizing at each step a quadratic approximation of f . In fact, we can write,

$$f(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T \nabla^2 f(x_k)s + \beta(x_k, s),$$

where $\beta(x_k, s)/\|s\|^2 \rightarrow 0$ for $s \rightarrow 0$. Thus, for small values of $\|s\|$, we can attempt to approximate $f(x_k + s)$ with the quadratic function

$$q_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2}s^T \nabla^2 f(x_k)s.$$

Then we can set $x_{k+1} = x_k + s_k$ and we can compute s_k by minimizing (if possible) the quadratic function $q_k(s)$ with respect to s .

As we have

$$\nabla q_k(s) = \nabla f(x_k) + \nabla^2 f(x_k)s,$$

if $\nabla^2 f(x_k)$ is positive definite the minimum point of $q_k(s)$ is given by

$$s_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k).$$

Thus Newton's method is defined by the iteration

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1}\nabla f(x_k). \quad (13.2)$$

It easily seen that if we identify the function F with the gradient ∇f of a function $f : R^n \rightarrow R$ then the Jacobian matrix of F can be identified with the Hessian matrix of f and hence (13.1) is equivalent to (13.2).

13.2 Local Convergence and Convergence Rate

We study here the local convergence of Newton's method for solving the system $F(x) = 0$ under the assumption that there exists $x^* \in R^n$ such that $F(x^*) = 0$.

Proposition 13.1 (Local Convergence of Newton's Method) *Let $F : R^n \rightarrow R^n$ be continuously differentiable on an open set $\mathcal{D} \subseteq R^n$. Suppose that there exists $x^* \in \mathcal{D}$ such that $F(x^*) = 0$ and that the Jacobian matrix $J(x^*)$ is non singular. Then there exists an open ball $\mathcal{B}(x^*; \varepsilon) \subseteq \mathcal{D}$, such that if $x_0 \in \mathcal{B}(x^*; \varepsilon)$, the sequence $\{x_k\}$ generated by Newton's method*

$$x_{k+1} = x_k - [J(x_k)]^{-1} F(x_k)$$

is well defined, $x_k \in \mathcal{B}(x^; \varepsilon)$ for all k and the sequence converges to x^* at a Q -superlinear convergence rate. Moreover if J is Lipschitz-continuous on \mathcal{D} , that is if there exists $L > 0$ such that*

$$\|J(x) - J(y)\| \leq L \|x - y\| \quad \text{for all } x, y \in \mathcal{D}$$

then the convergence rate is at least Q -quadratic.

Proof As $J(x^*)$ is non singular and J is continuous on \mathcal{D} , we can find $\varepsilon_1 > 0$ and $\mu > 0$ such that $\mathcal{B}(x^*; \varepsilon_1) \subseteq \mathcal{D}$ and that:

$$\left\| J(x)^{-1} \right\| \leq \mu, \quad \text{for all } x \in \mathcal{B}(x^*; \varepsilon_1).$$

Moreover, by continuity of J , for every $\sigma \in (0, 1)$ we can find $\varepsilon < \varepsilon_1$ such that

$$\|J(x) - J(y)\| \leq \sigma/\mu, \quad \text{for all } x, y \in \mathcal{B}(x^*; \varepsilon). \quad (13.3)$$

Suppose now that $x_k \in \mathcal{B}(x^*; \varepsilon)$. As $F(x^*) = 0$ by assumption, we can write Newton's iteration into the form::

$$x_{k+1} - x^* = -J(x_k)^{-1} [-J(x_k)(x_k - x^*) + F(x_k) - F(x^*)],$$

which implies:

$$\begin{aligned}\|x_{k+1} - x^*\| &\leq \|J(x_k)^{-1}\| \left\| -J(x_k)(x_k - x^*) + F(x_k) - F(x^*) \right\| \\ &\leq \mu \left\| -J(x_k)(x_k - x^*) + F(x_k) - F(x^*) \right\|.\end{aligned}\tag{13.4}$$

As F is differentiable, we can write

$$F(x_k) - F(x^*) = \int_0^1 J(x^* + \lambda(x_k - x^*))(x_k - x^*) d\lambda.$$

By substituting this expression into (13.4) we obtain

$$\|x_{k+1} - x^*\| \leq \mu \left\| \int_0^1 (J(x^* + \lambda(x_k - x^*)) - J(x_k))(x_k - x^*) d\lambda \right\|.$$

It follows

$$\|x_{k+1} - x^*\| \leq \mu \int_0^1 \|J(x^* + \lambda(x_k - x^*)) - J(x_k)\| d\lambda \|x_k - x^*\|,\tag{13.5}$$

and hence, by (13.3) and the convexity of $\mathcal{B}(x^*; \varepsilon)$, we obtain

$$\|x_{k+1} - x^*\| \leq \sigma \|x_k - x^*\|.\tag{13.6}$$

This implies $x_{k+1} \in \mathcal{B}(x^*, \varepsilon)$ and therefore, by induction, if $x_0 \in \mathcal{B}(x^*, \varepsilon)$ we have $x_k \in \mathcal{B}(x^*, \varepsilon)$ for all k . Then, using (13.6) we obtain $\|x_k - x^*\| \leq \sigma^k \|x_0 - x^*\|$, and hence, as $\sigma < 1$, we have that $\{x_k\}$ converges to x^* .

By (13.5), assuming that $x_k \neq x^*$ for all k , dividing by $\|x_k - x^*\|$ and taking limits for $k \rightarrow \infty$, we obtain, by the continuity of J :

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0,$$

which shows that the convergence rate is Q-superlinear. Finally, if J is Lipschitz-continuous from (13.5) we obtain

$$\|x_{k+1} - x^*\| \leq \mu L \int_0^1 (1-\lambda) d\lambda \|x_k - x^*\|^2 = \frac{\mu L}{2} \|x_k - x^*\|^2,\tag{13.7}$$

and hence the convergence rate is at least Q-quadratic. \square

Remark 13.1 An important remark is that the inversion of the Jacobian matrix is not requested in computations and would be unnecessarily expensive. The existence of the inverse must be requested in the theoretical analysis, however Newton's method

can be realized by determining the search direction through the solution of a linear system. In fact, the method could be conveniently described letting: $x_{k+1} = x_k + s_k$, where s_k is the solution of the system $J(x_k)s = -F(x_k)$. \square

Remark 13.2 A consequence of the preceding proposition is that if $\{x_k\}$ is any sequence (not necessarily produced by Newton's method in its pure form) converging to x^* , if s_k is Newton's direction for sufficiently large k and if the assumptions of Proposition 13.1 are satisfied then we have

$$\lim_{k \rightarrow \infty} \frac{\|x_k + s_k - x^*\|}{\|x_k - x^*\|} = 0.$$

In particular, if the Jacobian matrix J is Lipschitz continuous, we have also for large k that $\|x_k + s_k - x^*\| \leq C\|x_k - x^*\|^2$, for some $C > 0$. \square

The results given in Proposition 13.1 can be restated with reference to unconstrained optimization problems for computing a stationary point of an objective function $f : R^n \rightarrow R$ by identifying F with ∇f and J with $\nabla^2 f$.

Proposition 13.2 (Local Convergence of Newton's Method) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on an open set $\mathcal{D} \subseteq R^n$. Suppose that there exists $x^* \in \mathcal{D}$ such that $\nabla f(x^*) = 0$ and that the Hessian matrix $\nabla^2 f(x^*)$ is non singular. Then there exists an open ball $\mathcal{B}(x^*; \varepsilon) \subseteq \mathcal{D}$, such that if $x_0 \in \mathcal{B}(x^*; \varepsilon)$, the sequence $\{x_k\}$ generated by Newton's method*

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

is well defined, $x_k \in \mathcal{B}(x^; \varepsilon)$ for all k and the sequence converges to x^* at a Q-superlinear convergence rate. Moreover if $\nabla^2 f$ is Lipschitz-continuous on \mathcal{D} , that is if there exists $L > 0$ such that*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathcal{D}$$

then the convergence rate is at least Q-quadratic. \square

Recalling Remark 13.1, also in the case of function optimization, Newton's method can be defined by $x_{k+1} = x_k + s_k$, where s_k is a solution of the linear system $\nabla^2 f(x_k)s = -\nabla f(x_k)$.

Note also that the result of Proposition 13.2 is valid in a neighborhood of any stationary point (local minimum, local maximum, saddle point) where the Hessian matrix is nonsingular.

In the sequel we will refer essentially to optimization problems; the case of nonlinear equations will be studied in a Chap. 16.

13.3 Shamanskii Method

A modification of Newton's method, known as *Shamanskii method* [239], consists in updating the Jacobian matrix (or the Hessian matrix) not at each iteration as in Newton's method, but only periodically every $m + 1$ iterations with $m \geq 1$ ($m = 0$ will correspond to the standard method). The motivation is that of reducing the computational cost required for evaluating the derivatives and for solving the linear system, while retaining a good convergence rate.

In the sequel we consider the case of Newton's method for optimization problems and we restrict our analysis to the case $m = 1$. In this case Shamanskii method will be defined by the iteration

$$x_{k+1} = x_k^N - \nabla^2 f(x_k)^{-1} \nabla f(x_k^N), \quad (13.8)$$

where x_k^N is the point that one Newton's iteration would generate, that is:

$$x_k^N = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k).$$

Thus one iteration of Shamanskii method for $m = 1$ is constituted by two steps: the first step is exactly a Newton step, while the second step is a modified Newton step, where we use the same inverse Hessian employed in the first step.

A “true” two-steps Newton's method would be defined by

$$x_{k+1} = x_k^N - \nabla^2 f(x_k^N)^{-1} \nabla f(x_k^N). \quad (13.9)$$

When the assumptions of Proposition 13.2 are satisfied and the Hessian matrix is Lipschitz continuous, it obviously follows from this proposition that a two-step method defined by (13.9) would converge locally to a stationary point x^* with fourth order convergence rate at least, that is

$$\|x_{k+1} - x^*\| \leq C \|x_k - x^*\|^4, \text{ with } C > 0.$$

In the next proposition we show that, under the same assumptions, Shamanskii method with $m = 1$, defined by (13.8), converges locally at x^* at least with cubic convergence rate.

Proposition 13.3 (Local Convergence Of Shamanskii Method) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on an open set $\mathcal{D} \subseteq R^n$. Suppose that there exists $x^* \in \mathcal{D}$ such that $\nabla f(x^*) = 0$ and that the Hessian*

(continued)

Proposition 13.3 (continued)

matrix $\nabla^2 f(x^)$ is non singular. Suppose also that there exists $L > 0$ such that*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathcal{D}.$$

Then there exists an open ball $\mathcal{B}(x^; \varepsilon) \subseteq \mathcal{D}$, such that if $x_0 \in \mathcal{B}(x^*; \varepsilon)$, the sequence $\{x_k\}$ generated by Shamanskii method*

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

with $x_k^N = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$, is well defined, $x_k \in \mathcal{B}(x^; \varepsilon)$ for all k , the sequence converges to x^* and there exists $C > 0$, such that.*

$$\|x_{k+1} - x^*\| \leq C\|x_k - x^*\|^3. \quad (13.10)$$

Proof As $\nabla^2 f(x^*)$ is non singular and $\nabla^2 f$ is continuous on \mathcal{D} , we can find $\varepsilon_1 > 0$ and $\mu > 0$ such that $\mathcal{B}(x^*; \varepsilon_1) \subseteq \mathcal{D}$ and:

$$\|\nabla^2 f(x^*)\| \leq \mu, \quad \text{for all } x \in \mathcal{B}(x^*; \varepsilon_1).$$

It can be easily verified that, as $\nabla f(x^*) = 0$, taking into account definition of one iteration of the method, we can write:

$$\begin{aligned} x_{k+1} - x^* &= x_k^N - x^* - \nabla^2 f(x_k)^{-1} \nabla f(x_k^N) \\ &\quad + [\nabla^2 f(x_k)]^{-1} \nabla^2 f(x^*)(x_k^N - x^*) - [\nabla^2 f(x_k)]^{-1} \nabla^2 f(x^*)(x_k^N - x^*) \\ &= [-\nabla^2 f(x_k)]^{-1} [\nabla f(x_k^N) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k^N - x^*)] \\ &\quad + [\nabla^2 f(x_k)]^{-1} [\nabla^2 f(x_k)(x_k^N - x^*) - \nabla^2 f(x^*)(x_k^N - x^*)]. \end{aligned}$$

By the preceding equation, using the fact that

$$\begin{aligned} \nabla f(x_k^N) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k^N - x^*) &= \\ \int_0^1 [\nabla^2 f(x^* + \lambda(x_k^N - x^*)) - \nabla^2 f(x^*)](x_k^N - x^*) d\lambda, \end{aligned}$$

and taking into account the assumption of Lipschitz continuity of $\nabla^2 f$, we can write:

$$\|x_{k+1} - x^*\| \leq \frac{\eta L}{2} \|x_k^N - x^*\|^2 + \eta L \|x_k^N - x^*\| \|x_k - x^*\|. \quad (13.11)$$

On the other hand, by repeating the same reasoning used in the proof of Proposition 13.1 for establishing (13.7), we get

$$\|x_k^N - x^*\| \leq \frac{\eta L}{2} \|x_k - x^*\|^2. \quad (13.12)$$

By (13.11), assuming that $\|x_k - x^*\| < 1$ and recalling (13.12), we obtain

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \frac{\eta^3 L^3}{8} \|x_k - x^*\|^4 + \frac{\eta^2 L^2}{2} \|x_k - x^*\|^3 \\ &\leq \frac{\eta^3 L^3}{8} \|x_k - x^*\|^3 + \frac{\eta^2 L^2}{2} \|x_k - x^*\|^3 \\ &= \|x_k - x^*\|^2 \left(\frac{\eta^3 L^3}{8} + \frac{\eta^2 L^2}{2} \right) \|x_k - x^*\| \end{aligned} \quad (13.13)$$

Let now $\epsilon < \min\{1, \epsilon_1\}$ such that

$$\epsilon^2 \left(\frac{\eta^3 L^3}{8} + \frac{\eta^2 L^2}{2} \right) < 1,$$

and assume that $x_k \in \mathcal{B}(x^*; \epsilon)$.

From (13.13) it follows that $x_{k+1} \in \mathcal{B}(x^*; \epsilon)$ and hence, by induction, we have $x_k \in \mathcal{B}(x^*; \epsilon)$ for all k .

By repeated application of (13.13) we have also

$$\|x_k - x^*\| \leq \left(\epsilon^2 \frac{\eta^3 L^3}{8} + \epsilon^2 \frac{\eta^2 L^2}{2} \right)^k \|x_0 - x^*\|,$$

whence it follows, as $\left(\epsilon^2 \frac{\eta^3 L^3}{8} + \epsilon^2 \frac{\eta^2 L^2}{2} \right) < 1$, that $x_k \rightarrow x^*$. Then (13.13) implies that (13.10) holds. \square

In the general case, one iteration of Shamanskii method can be represented with the following scheme.

One Iteration of Shamanskii Method

- set $x_{k,0} = x_k$;
- For $i = 1, \dots, m + 1$:
 - set $x_{k,i} = x_{k,i-1} - [\nabla^2 f(x_k)]^{-1} \nabla f(x_{k,i-1})$;
 - End for
- set $x_{k+1} = x_{k,m+1}$.

When $m \geq 1$, under the assumptions of Proposition 13.3, it can be shown that Shamanskii method has convergence rate of order $m + 2$ at least.

13.4 Globalization of Newton's Method for Minimization

We consider in the sequel the globalization techniques used in the construction of Newton-type algorithms for the solution of unconstrained minimization problems with twice continuously differentiable objective function f .

In the general case, if we attempt to employ Newton's method in its pure form, starting from an arbitrary point $x_0 \in R^n$ we can incur in the following problems.

- (i) Newton's direction cannot be defined at x_k ($\nabla^2 f(x_k)$ is singular);
- (ii) the sequence produced by Newton's method does not admit limit points;
- (iii) no limit point is a stationary points of f ;
- (iv) Newton's direction could not be a descent direction and we may have limit points that are local maximum points of f .

In order to cope with these difficulties we must modify Newton's iteration, but we should retain, as much as possible, the local convergence properties of pure Newton's method.

The desirable features of a Newton-type algorithm are collected in the following definition.

Definition 13.1 (Globally Convergent Newton-Type Algorithm)

Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and assume that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact.

Consider the algorithm defined by $x_{k+1} = x_k + s_k$, with $\nabla f(x_k) \neq 0$ for all k . We say that it is a globally convergent Newton-type algorithm if it possesses the following properties

(continued)

Definition 13.1 (continued)

- (i) there exist limit points of $\{x_k\}$ and every limit point is a stationary point of f in \mathcal{L}_0 ;
- (ii) if $\{x_k\}_K$ is an infinite subsequence converging to a limit point, there exists at least an infinite subsequence of it such that the objective function is strictly decreasing;
- (iii) if $\{x_k\}$ converges towards a local minimum point x^* of f and $\nabla^2 f$ is a positive definite matrix that satisfies the assumptions of Proposition 13.2, then there exists k^* such that, for all $k \geq k^*$ we have

$$s_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

and hence the iteration of the algorithm is the pure Newton iteration. \square

It can be easily shown that if (ii) holds, then no limit point of $\{x_k\}$ is a local maximizer of the function f .

Globally convergent Newton type algorithms can be distinguished into:

- methods employing a line search along a descent direction obtained by modifying, when necessary, Newton's direction;
- trust region methods, where the new point x_{k+1} is obtained from the solution of a constrained problem, by minimizing a quadratic approximation (not necessarily convex) of f over a closed ball around x_k .

Trust region methods will be studied in Chap. 14. Thus, in the remaining sections of this chapter we will consider only Newton-type methods based on line searches, which constitute the basic approach to globalization of Newton's method. The iterations of these methods will be then of the form

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

where α_k is the step-size and d_k is the search direction obtained by modifying, if needed, Newton's direction.

Line search based techniques can be divided into two groups:

- *hybrid methods*, based on the combination of Newton's method with a globally convergent algorithm, such as the steepest descent method;
- *modification methods* where the Hessian matrix is modified, for instance, through the addition of a matrix constructed during a factorization process.

Another distinction is related to the line search, that is:

- *monotone methods* where the objective function is reduced at each iteration;

- *non monotone methods*, which admit controlled increases of the objective function during a finite number of iterations.

Non monotone methods will be studied in Chap. 24; here we will only refer to monotone methods.

In all cases, an important requirement for defining a globally convergent Newton-type algorithm in the sense of Definition 13.1 is that the step-size $\alpha = 1$ must be accepted in the line search, for sufficiently large k , if d_k is Newton's direction and the assumptions of Proposition 13.2 are satisfied.

In the next proposition, with reference to Armijo's method we indicate that the unit step-size can be actually accepted under the assumption stated.

Proposition 13.4 (Acceptance of Unit Step-Size) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and let $\{x_k\}$ be an infinite sequence produced by the algorithm*

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

where α_k is computed by Armijo's method. Suppose that the following conditions hold.

- (i) $\{x_k\}$ converges to x^* , where $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite;
- (ii) there exists an index \hat{k} such that, for all $k \geq \hat{k}$ the search direction d_k is Newton's direction, that is

$$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

Then, if $\gamma \in (0, 1/2)$, there exist an index $k^ \geq \hat{k}$, such that, for all $k \geq k^*$ we have*

$$f(x_k + d_k) \leq f(x_k) + \gamma \nabla f(x_k)^T d_k.$$

Proof As $\{x_k\}$ converges to x^* , by the assumptions made we can suppose that there exists some $k_1 \geq \hat{k}$ sufficiently large for the points x_k , with $k \geq k_1$ to remain in a closed ball around x^* where $\nabla^2 f(x_k)$ is positive definite and

$$\nabla^2 f(x_k) d_k = -\nabla f(x_k).$$

Then, for $k \geq k_1$ and some $\eta > 0$ we have:

$$-\nabla f(x_k)^T d_k = d_k^T \nabla^2 f(x_k) d_k \geq \eta \|d_k\|^2, \quad (13.14)$$

where η is a lower bound for the smallest eigenvalue of $\nabla^2 f(x)$ in the closed ball considered.

Using Taylor's theorem we can write:

$$f(x_k + d_k) = f(x_k) + \frac{1}{2} \nabla f(x_k)^T d_k + \frac{1}{2} \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T \nabla^2 f(z_k) d_k, \quad (13.15)$$

where $z_k = x_k + t_k d_k$ with $t_k \in (0, 1)$.

Recalling the first equality of (13.14) we have:

$$f(x_k + d_k) = f(x_k) + \frac{1}{2} \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T (\nabla^2 f(z_k) - \nabla^2 f(x_k)) d_k. \quad (13.16)$$

From (13.16), using again (13.14) we obtain:

$$\begin{aligned} f(x_k + d_k) - f(x_k) - \gamma \nabla f(x_k)^T d_k \\ = \left(\frac{1}{2} - \gamma \right) \nabla f(x_k)^T d_k + \frac{1}{2} d_k^T (\nabla^2 f(z_k) - \nabla^2 f(x_k)) d_k \\ \leq \left[-\left(\frac{1}{2} - \gamma \right) \eta + \frac{1}{2} \|\nabla^2 f(z_k) - \nabla^2 f(x_k)\| \right] \|d_k\|^2. \end{aligned}$$

Now, as $\nabla f(x_k) \rightarrow 0$ and $\nabla^2 f$ is non singular in the neighborhood considered, we have also that the Newton's direction d_k converges to zero and that z_k converges to x^* . Then, for sufficiently large k , if $\gamma < 1/2$, it follows from the preceding inequality that the term $\|\nabla^2 f(z_k) - \nabla^2 f(x_k)\|$ can be made arbitrarily small, so that we have

$$f(x_k + d_k) - f(x_k) - \gamma \nabla f(x_k)^T d_k \leq 0,$$

which completes the proof. \square

On the basis of these results, in order to construct a globally convergent Newton-type method, we must modify, when needed, the search direction in a way that some sufficient global convergence condition is satisfied and that the unit step-size is eventually accepted. These conditions should be automatically satisfied by Newton's direction, in a neighborhood of a minimum point where convergence of pure Newton's method can be established.

Recalling the results of Chaps. 9 and 10 a first step for the construction of a globalization scheme can be that of specifying, under usual assumptions, conditions that guarantee global convergence towards stationary points with an Armijo-type line search employing a unit initial step-size. We state the following proposition.

Proposition 13.5 (Conditions on Search Directions) *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be twice continuously differentiable on \mathbb{R}^n and assume that the level set \mathcal{L}_0 is compact. Suppose that $x_k \in \mathcal{L}_0$ for all k and that*

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^2, \quad c_1 > 0, \quad (13.17)$$

$$\|d_k\| \leq c_2 \|\nabla f(x_k)\|, \quad c_2 > 0. \quad (13.18)$$

Then, there exist forcing functions σ_1, σ_2 such that

$$\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \geq \sigma_1(\|\nabla f(x_k)\|), \quad (13.19)$$

$$\|d_k\| \geq \sigma_2 \left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right), \quad (13.20)$$

Proof It is easily seen that (13.17) and (13.18) imply that (13.19) holds, with

$$\sigma_1(t) = (c_1/c_2) t.$$

In order to establish (13.20), we can observe that (13.17) implies

$$\|\nabla f(x_k)\|^2 \leq \frac{1}{c_1} |\nabla f(x_k)^T d_k|. \quad (13.21)$$

On the other hand, using Schwarz inequality, we can write

$$\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \leq \|\nabla f(x_k)\| \quad (13.22)$$

and hence, using (13.22), (13.21) and again Schwarz inequality,, we have:

$$\left(\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right)^2 \leq \|\nabla f(x_k)\|^2 \leq \frac{1}{c_1} |\nabla f(x_k)^T d_k| \leq \frac{M}{c_1} \|d_k\|, \quad (13.23)$$

where M is an upper bound of $\|\nabla f(x)\|$ on \mathcal{L}_0 . Condition (13.20) will be then satisfied by taking $\sigma_2(t) = (c_1/M) t^2$. \square

We recall from Chap. 10 that (13.20) and Proposition 10.3 imply that Armijo's method with unit initial step-size guarantees that $\nabla f(x_k)^T d_k / \|d_k\| \rightarrow 0$. Therefore, by (13.19) and Proposition 9.2 we can establish convergence towards stationary points.

A first consequence of the preceding result is that the use of a line search along Newton's direction allows us to establish global convergence under strong convexity assumptions on the objective function. To show this, assume that the Hessian matrix (and hence also the inverse) is positive definite on R^n with uniformly bounded eigenvalues. To simplify notation, let

$$H(x) = \left(\nabla^2 f(x) \right)^{-1}$$

and suppose there exist $0 < m \leq M$ such that, for all $x \in R^n$ we have

$$M \geq \lambda_{\max}(H(x)) \geq \lambda_{\min}(H(x)) \geq m > 0.$$

Then it can be easily verified that the assumptions of the preceding proposition are satisfied. In fact, letting $H_k = H(x_k)$ we have

$$\begin{aligned} \nabla f(x_k)^T d_k &= -\nabla f(x_k)^T H_k \nabla f(x_k) \leq -\lambda_{\min}(H_k) \|\nabla f(x_k)\|^2 \leq -m \|\nabla f(x_k)\|^2 \\ \|d_k\| &= \|H_k \nabla f(x_k)\| \leq \lambda_{\max}(H_k) \|\nabla f(x_k)\| \leq M \|\nabla f(x_k)\|, \end{aligned}$$

Then, as noted before, the algorithm

$$x_{k+1} = x_k - \alpha_k \left(\nabla^2 f(x_k) \right)^{-1} \nabla f(x_k),$$

where α_k is computed by means of Armijo's method, with $\gamma < 1/2$ and $\Delta_k = 1$, will be a globally convergent Newton-type algorithm.

In the general case, in order to define a Newton-type algorithm in the sense of Definition 13.1 we must also show that the convergence conditions on the search direction are eventually satisfied by Newton's direction when we are converging to a minimum point satisfying the assumptions of Proposition 13.2. Thus, it could be useful to modify the convergence conditions stated above, in a way that these conditions become automatically less demanding as $\nabla f(x_k)$ converges to zero, but still enforce global convergence.

Let us define the function

$$\psi(x) = \min \{1, \|\nabla f(x)\|\}, \quad (13.24)$$

then we can modify the conditions considered in Proposition 13.5 in the following form

$$\nabla f(x_k)^T d_k \leq -c_1 \psi(x_k) \|\nabla f(x_k)\|^2, \quad c_1 > 0 \quad (13.25)$$

$$\psi(x_k) \|d_k\| \leq c_2 \|\nabla f(x_k)\|, \quad c_2 > 0. \quad (13.26)$$

13.5 Hybrid Methods

The simplest idea for constructing a globally convergent Newton-type algorithm can be that of employing the steepest descent direction whenever Newton's direction cannot be computed or it does not satisfy the convergence conditions. Then a line search algorithm can be employed for computing the step-size α_k .

An example of hybrid method can be the following scheme where we assume that the line search algorithm is an Armijo-type method with $\gamma < 1/2$ and initial step-size $\Delta_k = 1$.

Hybrid Newton-Type Method (HN)

Choose $x_0 \in R^n$.

For k=0,1,...

1. Compute $\nabla f(x_k)$; if $\nabla f(x_k) = 0$ terminate; otherwise compute $\nabla^2 f(x_k)$.
2. If the system: $\nabla^2 f(x_k)s = -\nabla f(x_k)$ has no solution then set $d_k = -\nabla f(x_k)$;
otherwise:
 - (i) compute a solution s^N and set $\psi_k = \min \{1, \|\nabla f(x_k)\|\}$,
 - (ii) If $|\nabla f(x_k)^T s^N| < c_1 \psi_k \|\nabla f(x_k)\|$ or $\psi_k \|s^N\| > c_2 \|\nabla f(x_k)\|$ then set $d_k = -\nabla f(x_k)$;
otherwise, if $\nabla f(x_k)^T s^N < 0$ set $d_k = s^N$ else set $d_k = -s^N$.
3. Compute α_k through the line search algorithm;
4. Set $x_{k+1} = x_k + \alpha_k d_k$.

End for

Remark 13.3 The criterion used for establishing the existence of solutions of the system $\nabla^2 f(x_k)s = -\nabla f(x_k)$, will depend, in practice, on the solution technique employed. We note that global convergence is guaranteed, in any case, by the conditions imposed on the search direction even if $\nabla^2 f(x_k)$ is singular or the solution of the system is not exact. However, consistency with Newton's method requires, in principle, that $\nabla^2 f(x_k)$ is non singular and that the solution of the system is exact. Inexact Newton-type methods will be discussed in the sequel. \square

Remark 13.4 The choice $d_k = -s^N$ at step 2(ii) when $\nabla f(x_k)^T s^N > 0$, is motivated by the fact that the direction $d_k = -s^N$, where s^N is Newton's direction, is both a descent direction and also a *negative curvature direction*. In fact, under the assumptions stated, we can easily verify that

$$d_k^T \nabla^2 f(x_k) d_k = \nabla f(x_k)^T [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) < 0.$$

A negative curvature direction can be quite promising, since the directional derivative is decreasing along this direction and hence a good reduction of the objective function could be expected. It could be convenient in this case the use of a line search technique that admits also extensions of the step-size. \square

The convergence of hybrid algorithm defined above is demonstrated in the next proposition.

Proposition 13.6 (Convergence of a Hybrid Algorithm) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and assume that the level set \mathcal{L}_0 is compact. Then algorithm (HN) is a globally convergent Newton-type algorithm in the sense of Definition 13.1.*

Proof We can assume that the algorithm does not terminate. Then it guarantees that the search direction d_k is a descent direction satisfying $\nabla f(x_k)^T d_k < 0$ and that we have $f(x_{k+1}) < f(x_k)$ for all k , so that $x_k \in \mathcal{L}_0$. This establishes (ii) of Definition 13.1. Moreover, if d_k is not the negative gradient direction, we have

$$|\nabla f(x_k)^T d_k| \geq c_1 \psi(x_k) \|\nabla f(x_k)\|^2 \quad (13.27)$$

and

$$\psi(x_k) \|d_k\| \leq c_2 \|\nabla f(x_k)\|, \quad (13.28)$$

where

$$\psi(x_k) = \min \{1, \|\nabla f(x_k)\|\}.$$

Now we show that (i) of Definition 13.1 holds, that is that the limit points are stationary points. Reasoning by contradiction, if the assertion is false there must exist a subsequence, which we redefine as $\{x_k\}$, and a number $\varepsilon > 0$, such that $\|\nabla f(x_k)\| \geq \varepsilon$ for all k . Taking into account the possibility that $d_k = -\nabla f(x_k)$, we have

$$|\nabla f(x_k)^T d_k| \geq \tilde{c}_1 \|\nabla f(x_k)\|^2 \quad (13.29)$$

and

$$\|d_k\| \leq \tilde{c}_2 \|\nabla f(x_k)\|, \quad (13.30)$$

where $\tilde{c}_1 = \min \{1, c_1 \min\{1, \varepsilon\}\}$ and $\tilde{c}_2 = \max \{1, c_2 / \min\{1, \varepsilon\}\}$. Under the assumption stated, we have that (13.29) (13.30) imply that the assumptions of

Proposition 13.5 are satisfied, and hence, assuming unit initial step-size in the line search, recalling Remark 10.1 (in the subsequence considered) we have that

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0. \quad (13.31)$$

By Proposition 13.5 we have that d_k satisfies also the assumptions of Proposition 9.2 and hence we get a contradiction. Thus, also (i) of Proposition 13.1 is satisfied.

Finally, we must show that condition (iii) of Definition 13.1 holds. Assume then that $\{x_k\}$ converges towards a local minimum point x^* such that $\nabla^2 f(x^*)$ is positive definite and that the assumptions of Proposition 13.2 hold. We show that the search direction d_k is Newton's direction for sufficiently large values of k .

As $\nabla^2 f(x^*)$ is positive definite, $\nabla^2 f$ is continuous and $x_k \rightarrow x^*$, there must exist k_1 such that for $k \geq k_1$ the matrix $\nabla^2 f(x_k)$ is positive definite and hence the algorithm computes $s_k^N = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$. Moreover, we have:

$$|\nabla f(x_k)^T s_k^N| = \nabla f(x_k)^T [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \geq m \|\nabla f(x_k)\|^2,$$

where $m > 0$ is a lower bound of the minimum eigenvalue of $[\nabla^2 f]^{-1}$ in a neighborhood of x^* . As $\nabla f(x_k) \rightarrow 0$ and hence $\psi(x_k) \rightarrow 0$, there must exist $k_2 \geq k_1$ such that, for $k \geq k_2$ condition (13.27) is satisfied. Moreover, for sufficiently large values of k we have

$$\|s_k^N\| = \|\nabla f(x_k)^T [\nabla^2 f(x_k)]^{-2} \nabla f(x_k)\|^{1/2} \leq M \|\nabla f(x_k)\|,$$

where $M > 0$ is an upper bound of the maximum eigenvalue of $[\nabla^2 f]^{-1}$ in the same neighborhood of x^* considered above. As $\nabla f(x_k) \rightarrow 0$, there must exist $k_3 \geq k_2$ such that for $k \geq k_3$ also condition (13.28) is satisfied.

It can be concluded that, for $k \geq k_3$, d_k is Newton's direction and hence, by Proposition 13.2 the initial step-size $\alpha_k = 1$ is accepted in Armijo's method for sufficiently large k (say $k \geq k^* \geq k_3$). Then for $k \geq k^*$ we will have

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

and hence the algorithm is a globally convergent Newton-type algorithm in the sense of Definition 13.1. \square

13.6 Modification Methods

A globally convergent modification of Newton's method can be obtained by modifying the Hessian matrix in a way that the search direction is a descent direction, so that we can employ a line search algorithm that guarantees global convergence.

In particular, we can define a symmetric matrix E such that $\nabla^2 f(x_k) + E_k$ is “sufficiently” definite positive. In this case we can define

$$d_k = -\left[\nabla^2 f(x_k) + E_k\right]^{-1} \nabla f(x_k),$$

and we can perform a line search along d_k .

A class of methods for constructing E_k is that of methods employing a *modified Cholesky factorization* technique, which determines a *diagonal matrix* E_k and a *lower triangular matrix* L_k with positive diagonal elements l_{ii} , such that:

$$\nabla^2 f(x_k) + E_k = L_k L_k^T.$$

Alternatively, we can also construct a positive definite diagonal matrix D_k such that

$$\nabla^2 f(x_k) + E_k = L_k D_k L_k^T.$$

We illustrate a modification technique in more detail with reference to the LL^T factorization. First we recall the LL^T Cholesky factorization of a positive definite matrix.

Let A be a symmetric positive definite matrix. The Cholesky factorization of A can be computed by solving the equation $A = LL^T$. We can construct L by columns, noting that from the preceding equation we get, for $i \geq j$:

$$a_{ij} = \sum_{k=1}^j l_{ik} l_{jk}.$$

Letting

$$l_{11} = \sqrt{a_{11}}$$

we have

$$l_{i1} = \frac{a_{i1}}{l_{11}} \quad i = 2, \dots, n,$$

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2}, \quad j = 2, \dots, n,$$

and, for $j = 2, \dots, n-1$

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}, \quad i = j+1, \dots, n.$$

We note that, if the matrix A is not positive definite, the preceding formulas cannot be used, as we should compute the square root of a negative number or perform a division by zero.

Consider now a modification technique for Newton's method; in order to simplify notation we set provisionally:

$$A = \nabla^2 f(x_k).$$

It can be observed that the factorization defined above can be performed by adding positive numbers δ_j to the diagonal elements.

In particular, we can construct the first column of L by choosing $\delta_1 \geq 0$ and $\varepsilon_1 > 0$ such that $a_{11} + \delta_1 \geq \varepsilon_1$ and then setting

$$l_{11} = \sqrt{a_{11} + \delta_1} \quad l_{i1} = \frac{a_{i1}}{l_{11}}, \quad i = 2, \dots, n.$$

Given the columns $1, 2, \dots, j-1$ of L , the j -th column can be defined by modifying the diagonal element by choosing $\delta_j \geq 0$ such that

$$a_{jj} + \delta_j \geq \varepsilon_j + \sum_{k=1}^{j-1} l_{jk}^2,$$

with $\varepsilon_j > 0$. Then we can set for $j = 2, \dots, n$:

$$l_{jj} = \sqrt{a_{jj} + \delta_j - \sum_{k=1}^{j-1} l_{jk}^2},$$

and for $j = 2, n-1$:

$$l_{ij} = \frac{a_{ij} - \sum_{k=1}^{j-1} l_{jk} l_{ik}}{l_{jj}}, \quad i = j+1, \dots, n.$$

The choice of the parameters ε_j and δ_j is not a trivial task, as there are, in general, some possibly contrasting requirements, as indicated below.

- Newton's method should be altered as little as possible; in particular, in a neighborhood of a local minimum point where the assumptions of Definition 13.1 are satisfied, we would like, in principle, that the Hessian matrix is not modified and hence that $\delta_j = 0$ for all j . In particular, the values the parameters ε_j could be made dependent on the gradient norm.
- Global convergence should be ensured, by choosing, for instance, the parameters in a way that some sufficient convergence condition is satisfied.

- The modified Hessian matrix should be well conditioned and hence too small diagonal elements should be avoided.
- The computational cost of the modified factorization technique should not be much higher than that of the Cholesky algorithm.

With reference to the LL^T factorization of $A = \nabla^2 f(x_k)$, once that the factor L has been computed, then we must solve the system:

$$LL^T d = -\nabla f(x_k).$$

The solution can be obtained solving first (by forward elimination) the triangular system

$$Ls = -\nabla f(x_k),$$

and then (by backward elimination) the triangular system

$$L^T d = s.$$

The solutions of the two systems are obtained by assuming

$$\begin{aligned} s_1 &= -\frac{\partial f(x_k)/\partial x_1}{l_{11}}, \\ s_i &= -\frac{\partial f(x_k)/\partial x_i + \sum_{k=1}^{i-1} l_{ik} s_k}{l_{ii}}, \quad i = 2, \dots, n, \end{aligned}$$

and then computing the components of the search direction:

$$d_n = \frac{s_n}{l_{nn}},$$

$$d_i = \frac{s_i - \sum_{k=i+1}^n l_{ki} d_k}{l_{ii}}, \quad i = n-1, \dots, 1.$$

Along the search direction d we can then perform a line search, by employing Armijo's method with initial step-size $\Delta_k = 1$.

13.7 Exercises

13.1 Define a computer code based on a hybrid modification of Newton's method and study the behavior of the method for a set of test problems.

13.2 Define a computer code based on a modified Cholesky factorization of Newton's method and study the behavior of the method for a set of test problems.

13.8 Notes and References

Local convergence properties of Newton's method were originally analyzed in [148]. The modification of Newton's method, known as Shamanskii method, has been proposed in [239]. A globally convergent version of the Shamanskii method has been presented in [161]. The globalization technique of Newton's method based on the modified Cholesky factorization has been introduced in [113]. In order to improve the computational aspects and to reduce the effect of numerical errors various different factorizations have been proposed [88].

Chapter 14

Trust Region Methods



In this chapter we present “trust region” methods, that is, methods where the search direction and the length of each step are simultaneously computed by minimizing a (possibly nonconvex) quadratic model of the objective function over a suitable neighborhood of the current point. Global convergence results are reported and methods for the solution of the constrained quadratic subproblem are described. We present a globalization strategy for Newton’s method based on the trust region approach. Finally, in connection with complexity analysis issues, we briefly present a class of adaptive regularized methods recently studied as an alternative to classical globalization techniques of Newton’s method.

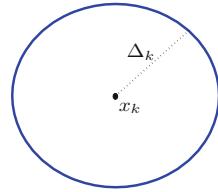
14.1 The General Trust Region Framework and Convergence Results

Consider an unconstrained minimization problem with a smooth objective function $f : R^n \rightarrow R$. The basic idea of “trust region” methods is that of determining at each iteration k , both the search direction and the length of the step by minimizing a quadratic model of the objective function over a (usually) spherical region around the current point x_k . In order to explain the motivation of the trust region-based approach, we recall that the search direction of the Newton method can be determined by minimizing the following quadratic approximation of f :

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s,$$

provided that the Hessian matrix is positive definite. However, if $\nabla^2 f(x_k)$ is not (at least) positive semidefinite then the function $m_k(s)$ does not admit a minimum.

Fig. 14.1 Trust region \mathcal{B}_k around x_k



In order to take into account this issue, a suitable strategy could be that of performing the minimization of $m_k(s)$ on a neighborhood of zero, that is, by considering the constraint $\|s\| \leq \Delta_k$, and setting

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

where $s_k \in R^n$ is a solution of the subproblem

$$\begin{aligned} \min m_k(s) &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s. \\ \|s\| &\leq \Delta_k \end{aligned} \quad (14.1)$$

The radius Δ_k defining the spherical region around x_k is usually determined in such a way that $f(x_{k+1}) < f(x_k)$ and that the reduction of f is close to that of the quadratic model $m_k(s)$. In this way, the radius defines the region where the model can be considered reliable, i.e., the so-called *trust region* (Fig. 14.1).

Before to formally define the scheme of a trust region method, we present the key elements. Let x_k be the current point.

- A quadratic model of the objective function is defined as follows

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s, \quad (14.2)$$

where B_k is a $n \times n$ symmetric matrix (note that B_k is neither necessarily equal to the Hessian matrix $\nabla^2 f(x_k)$ nor positive semidefinite).

- The radius $\Delta_k > 0$ of the trust region

$$\mathcal{B}_k = \{x \in R^n : \|x - x_k\| \leq \Delta_k\},$$

is defined.

- A (possibly approximate) solution s_k of the subproblem

$$\begin{aligned} \min m_k(s) &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \\ \|s\| &\leq \Delta_k \end{aligned} \quad (14.3)$$

is computed.

- The trial point $x_k + s_k$ is accepted as the new iterate x_{k+1} if s_k determines a “sufficient reduction” of the quadratic model $m_k(s)$ and if this reduction implies also a sufficient reduction of the true objective function f .
- If the trial point $x_k + s_k$ is accepted then the radius of the trust region can be increased (or held constant), otherwise it is necessarily reduced.

A formal scheme of a trust region method [50] is reported below.

Algorithm TR: Conceptual Model of a Trust Region Method

Data: Starting point $x_0 \in R^n$ and initial radius $\Delta_0 > 0$; positive constants $\eta_1, \eta_2, \gamma_1, \gamma_2$ such that

$$0 < \eta_1 < \eta_2 < 1 \quad \text{and} \quad 0 < \gamma_1 \leq \gamma_2 < 1. \quad (14.4)$$

Set $k = 0$.

While $\nabla f(x_k) \neq 0$ **do:**

Step 1: definition of the quadratic model.

Define a quadratic model as in (14.2).

Step 2: solution of the subproblem.

Compute a solution (possibly approximate) s_k of the subproblem (14.3) in such a way that a “sufficient reduction” $m_k(s_k) - m_k(0)$ of the quadratic model is attained.

Step 3: acceptance of the trial point.

Compute $f(x_k + s_k)$ and set

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}. \quad (14.5)$$

If $\rho_k \geq \eta_1$ then set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 4: updating of the radius.

Define the new radius Δ_{k+1} as follows

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (14.6)$$

(continued)

Set $k = k + 1$

End While

In the above scheme some instructions are partially unspecified (in particular, at Step 2):

- the meaning of *sufficient reduction* of the quadratic model;
- how to determine a (possibly approximate) solution s_k of the subproblem (14.3) to guarantee the sufficient reduction of the quadratic model.

We observe that the criterion defining the sufficient reduction of the quadratic model plays a crucial role to guarantee the global convergence of the algorithm.

First we state below the formal definition of a condition of sufficient reduction.

Assumption 14.1 We say that the vector s_k computed at Step 2 satisfies the condition of sufficient reduction of the quadratic model if

$$m_k(0) - m_k(s_k) \geq c_1 \|\nabla f(x_k)\| \min \left(\Delta_k, \frac{\|\nabla f(x_k)\|}{1 + \|B_k\|} \right), \quad (14.7)$$

where $c_1 > 0$.

Note that (14.7) and the condition $\nabla f(x_k) \neq 0$ imply

$$m_k(s_k) < m_k(0).$$

In the next section we will present and discuss inexact and exact techniques to solve the trust region subproblem ensuring that the above sufficient reduction condition holds.

Now we state global convergence properties of the sequence generated by the trust region algorithm. The proofs of the convergence results can be found, for instance, in [50]. However, for completeness, these proofs are reported in the appendix of the chapter. First we state a weak convergence result.

Proposition 14.1 Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that f is bounded below. Let $\{x_k\}$ be the sequence generated by Algorithm TR assuming that Condition (14.1) holds and that there exists a

(continued)

Proposition 14.1 (continued)
constant β such that $\|B_k\| \leq \beta$ for all k . Then

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (14.8)$$

□

A stronger convergence result can be established by assuming the Lipschitz continuity of the gradient.

Proposition 14.2 *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that f is bounded below. Let $\{x_k\}$ be the sequence generated by Algorithm TR assuming that Condition (14.1) holds and that there exists a constant β such that $\|B_k\| \leq \beta$ for all k . Assume that the gradient is Lipschitz-continuous on the level set \mathcal{L}_0 , i.e., there exists a constant $L > 0$ such that*

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in \mathcal{L}_0. \quad (14.9)$$

Then

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (14.10)$$

□

14.2 Classification of Solution Methods for the Trust Region Subproblem

A trust region algorithm is characterized by the method for computing a (possibly approximate) solution of the quadratic subproblem (14.3).

We distinguish between:

- (a) methods for computing an approximate solution;
- (b) methods for computing the exact solution.

The simplest method among those of class (a) is the method based on the Cauchy step along the steepest descent direction. We will show that the adoption of this method for approximately solving the trust region problem ensures that condition (14.1) holds and hence permits to guarantee global convergence properties

of the trust region algorithm. We also present other two methods of class (a), the *dogleg method* and the *conjugate gradient method of Steihaug*, designed to improve the efficiency with respect to the simple Cauchy step-based method. These methods guarantee also the global convergence of the trust region algorithm since they ensure a decrease of the quadratic model greater or equal than that obtained by the Cauchy step.

The methods of class (b) usually require suitable factorizations of the matrix B_k , so that, they can be advantageously applied whenever the number n of variables is not too high. The definition of methods for computing the exact solution depends on the fact that it is possible to state necessary and sufficient optimality conditions of global minima for subproblem (14.3). These conditions are related to the particular structure of the subproblem and will be analyzed in Sect. 14.6.

Note that we do not impose convexity assumptions on the objective function of subproblem (14.3), so that, even from a theoretical point of view, the existence of necessary and sufficient global optimality conditions for a nonconvex problem is quite important.

The analysis of methods for computing the exact solution of the trust region subproblem can be found, for instance, in [196]. We limit ourselves to describe three methods for computing an approximate solution.

14.3 The Cauchy Step-Based Method

Given the quadratic model

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s, \quad (14.11)$$

and assuming that $\nabla f(x_k) \neq 0$, we consider the problem of determining a vector s_k satisfying the trust region constraint $\|s_k\| \leq \Delta_k$ and ensuring a sufficient reduction $m_k(s_k) - m_k(0)$. The simplest technique is that of performing a step along the steepest descent direction starting from the point $s = 0$, that is, a step along the opposite gradient direction $-\nabla m_k(0) = -\nabla f(x_k)$. Formally, setting $s = -\tau \nabla f(x_k)$ in (14.1), we obtain the one dimensional problem

$$\begin{aligned} \min \quad & m_k(\tau) = f(x_k) - \tau \|\nabla f(x_k)\|^2 + \frac{1}{2} \tau^2 \nabla f(x_k)^T B_k \nabla f(x_k) \\ \text{s.t. } & 0 \leq \tau \leq \frac{\Delta_k}{\|\nabla f(x_k)\|}. \end{aligned} \quad (14.12)$$

In order to compute the solution τ^* of (14.12) we consider the two exhaustive cases.

Case I $\nabla f(x_k)^T B_k \nabla f(x_k) \leq 0$. In this case the objective function is strictly monotonically decreasing for increasing values of $\tau > 0$, so that, the minimizer is the upper bound of the interval defining the feasible set, and hence we have

$$\tau^* = \frac{\Delta_k}{\|\nabla f(x_k)\|}. \quad (14.13)$$

Case II $\nabla f(x_k)^T B_k \nabla f(x_k) > 0$. In this case the objective function is a strictly convex quadratic function. From the optimality conditions of box constrained problems it follows

$$\tau^* = \min \left\{ \frac{\Delta_k}{\|\nabla f(x_k)\|}, \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)} \right\}. \quad (14.14)$$

We observe that the point

$$\frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)}$$

is the unconstrained global minimizer of $m_k(\tau)$, so formula (14.14) takes into account the constraint $\tau \leq \frac{\Delta_k}{\|\nabla f(x_k)\|}$.

We define *Cauchy step* the direction

$$s_k^c = -\tau^* \nabla f(x_k), \quad (14.15)$$

where

$$\tau^* = \begin{cases} \frac{\Delta_k}{\|\nabla f(x_k)\|} & \text{if } \nabla f(x_k)^T B_k \nabla f(x_k) \leq 0 \\ \min \left\{ \frac{\Delta_k}{\|\nabla f(x_k)\|}, \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T B_k \nabla f(x_k)} \right\} & \text{if } \nabla f(x_k)^T B_k \nabla f(x_k) > 0. \end{cases} \quad (14.16)$$

We define *Cauchy point* the point

$$x_k^c = x_k + s_k^c = x_k - \tau^* \nabla f(x_k).$$

As reported in the next proposition [50], the Cauchy step s_k^c satisfies the condition of sufficient reduction of the quadratic model stated in (14.1).

Proposition 14.3 *The Cauchy step s_k^c defined by (14.15) satisfies condition (14.1) with $c_1 = 1/2$, that is*

$$m_k(0) - m_k(s_k^c) \geq \frac{1}{2} \|\nabla f(x_k)\| \min \left(\Delta_k, \frac{\|\nabla f(x_k)\|}{1 + \|B_k\|} \right). \quad (14.17)$$

□

14.4 The Dogleg Method

Let us consider the problem (14.3) rewritten as follows in order to simplify notation (we omit the dependence on k , and we indicate by g the gradient of f computed at x_k)

$$\begin{aligned} \min_{s \in R^n} m(s) &= f + g^T s + \frac{1}{2} s^T B s \\ \|s\| &\leq \Delta. \end{aligned} \quad (14.18)$$

Suppose that the matrix B is symmetric and positive definite. The solution of subproblem (14.18) depends on the radius Δ of the trust region. Roughly speaking, if Δ is “sufficiently large” then the solution is the unconstrained minimizer of the quadratic function, i.e.,

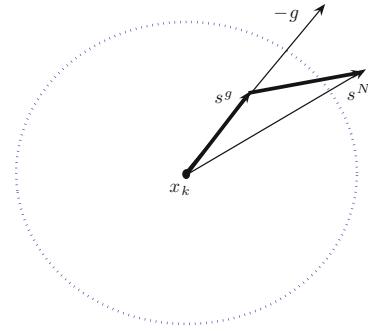
$$s^*(\Delta) = -B^{-1}g. \quad (14.19)$$

Viceversa, if the radius Δ is “sufficiently small” then the solution does not strongly depends on the quadratic term and is approximately equal to that of the affine model $f + g^T s$, i.e.,

$$s^*(\lambda) \approx -\Delta \frac{g}{\|g\|}. \quad (14.20)$$

For intermediate values of Δ the solution $s^*(\Delta)$ will follow a curvilinear trajectory. The idea underlying the *dogleg method* is that of approximating the curvilinear trajectory by two line segments. The first segment joins the origin and the minimum point of the quadratic model along the steepest direction, and it is defined by the vector

$$s^g = -\frac{g^T g}{g^T B g} g. \quad (14.21)$$

Fig. 14.2 Dogleg path $\tilde{s}(\tau)$ 

The second segment joins the vector s^g and the point defined by (14.19) and denoted by $s^N = -B^{-1}g$. The dogleg trajectory is presented in Fig. 14.2. Formally we can define the trajectory as follows

$$\tilde{s}(\tau) = \begin{cases} \tau s^g & 0 \leq \tau \leq 1 \\ s^g + (\tau - 1)(s^N - s^g) & 1 \leq \tau \leq 2 \end{cases} \quad (14.22)$$

Before to describe the method we state the following result [196].

Lemma 14.1 Consider the quadratic model

$$m(s) = f + g^T s + \frac{1}{2} s^T B s,$$

where the matrix B is symmetric and positive definite. Then

- (i) $\|\tilde{s}(\tau)\|$ is an increasing function di τ ;
- (ii) $m(\tilde{s}(\tau))$ is a decreasing function of τ .

□

It is easy to see that the assertions are true for $\tau \in [0, 1]$. As regards the case $\tau \in [1, 2]$, in order to prove assertion (i), the function

$$h(\tau) = \frac{1}{2} \|\tilde{s}(\tau)\|^2 = \frac{1}{2} \|s^g + (\tau - 1)(s^N - s^g)\|^2,$$

is considered and it is shown that $h'(\tau) \geq 0$. To prove assertion (ii), as $m(s)$ is a strictly convex function over R^n under the stated assumptions, we have that $s^* = -B^{-1}g = \tilde{s}(2)$ is the unique global minimizer. Then, the function of one variable $m(\tau) = m(\tilde{s}(\tau))$ is a strictly convex function whose unique global minimizer is $\tau = 2$. By exploiting the fact that $m(\tau)$ is convex and differentiable, it is possible to show that $m'(\tau) < 0$ for every $\tau \in [1, 2]$.

From the above lemma we get that the objective function is a decreasing function along $\tilde{s}(\tau)$ and this leads to summarize the method as follows:

- (i) if $\|s^N\| \leq \Delta$ then set $s^* = s^N$;
- (ii) if $\|s^N\| > \Delta$ then

- (iia) if $\|s^g\| \leq \Delta$ then determine the value $\tau^* \in [1, 2]$ such that

$$\|s^g + (\tau^* - 1)(s^N - s^g)\|^2 = \Delta^2,$$

and set $s^* = s^g + \tau^*(s^N - s^g)$;

- (iib) if $\|s^g\| > \Delta$ then $s^* = -\Delta \frac{g}{\|g\|}$.

Note that in all the cases we have a decrease of the quadratic model greater or equal than that obtained by the Cauchy step. Indeed, in case (i) we have

$$m(s^N) \leq m(s^c)$$

since s^N is the global minimizer of the quadratic function.

In case (iia) we have $m(s^*) \leq m(s^g) \leq m(s^c)$. In case (iib) we have $s^* = s^c$.

14.5 The Conjugate Gradient Method of Steihaug

The method is a modified version of the standard conjugate gradient method for computing an approximate solution of the trust region subproblem (14.18), which is a quadratic problem (not necessarily convex) with a spherical constraint.

The modifications of the conjugate gradient method mainly concern the stopping criterion. More specifically, further stopping criteria are added and are related to:

- the violation of the trust region constraint;
- the generation of a direction with *negative curvature*, that is, a direction d such that $d^T Bd \leq 0$.

The method can be formally described as follows.

The Conjugate Gradient Method of Steihaug

Data: $\epsilon > 0$.

1. Set $s_0 = 0$, $r_0 = g$, $d_0 = -g$, $j = 0$.
2. If $\|r_0\| \leq \epsilon$ then $s = s_j$ and exit.
3. If $d_j^T Bd_j \leq 0$

(continued)

determine the positive value τ^* such that

$$\|s_j + \tau^* d_j\| = \Delta,$$

- set $s = s_j + \tau^* d_j$ and exit.
 4. Set $\alpha_j = r_j^T r_j / d_j^T B d_j$, $s_{j+1} = s_j + \alpha_j d_j$.
 5. If $\|s_{j+1}\| \geq \Delta$ then determine the positive value τ^* such that

$$\|s_j + \tau^* d_j\| = \Delta,$$

- set $s = s_j + \tau^* d_j$ and exit.
 6. Set $r_{j+1} = r_j + \alpha_j B d_j$. If $\|r_{j+1}\| \leq \epsilon$ then set $s = s_{j+1}$ and exit.
 7. Set $\beta_{j+1} = r_{j+1}^T r_{j+1} / r_j^T r_j$, $d_{j+1} = -r_{j+1} + \beta_{j+1} d_j$, $j = j + 1$ and go to Step 3.

As already said, the above scheme is different from the standard conjugate gradient method in the stopping criteria introduced at Step 3 and Step 5.

We remark that the initialization $s_0 = 0$ is fundamental from a theoretical point of view to ensure global convergence properties of the trust region method. Indeed, from the instructions of the conjugate gradient method and from the initialization $s_0 = 0$ it follows

$$s_1 = \alpha_0 d_0 = \frac{r_0^T r_0}{d_0^T B d_0} d_0 = -\frac{-g^T g}{g^T B g} g.$$

If either $d_0^T B d_0 \leq 0$ or $\|s_1\| \geq \Delta$, then the method terminates at Step 3 and Step 5 respectively, providing the direction

$$s = \frac{\Delta}{\|g\|} g.$$

Therefore, with or without the stopping criteria, the first iteration always provides the Cauchy step. As already seen, at each iteration of the conjugate gradient method the quadratic function decreases. Hence, the direction s obtained by the above scheme determines a decrease of the quadratic model greater or equal than that obtained by the Cauchy step, and this is sufficient to guarantee global convergence properties of the trust region method.

Finally, as stated in the proposition reported below, thanks to the initialization $s_0 = 0$, the method generates vectors s_j with increasing norm. Then, the stopping criterion of Step 5 is well-motivated, since the method can not provide a further decrease of the quadratic function without violating the trust region constraint.

Proposition 14.4 *The sequence $\{s_0, s_1, s_2, \dots, s_N\}$ generated by the conjugate gradient method is such that*

$$0 = \|s_0\| < \|s_1\| < \dots \|s_j\| < \|s_{j+1}\| < \dots < \|s_N\| = \|s\| \leq \Delta. \quad (14.23)$$

□

14.6 Necessary and Sufficient Optimality Conditions for the Trust Region Subproblem

Let us consider again subproblem (14.3) rewritten in the form (14.18) to simplify the notation. We can state the following result.

Proposition 14.5 (Necessary Conditions of Local Minimum) *Let s^* be a local minimizer of problem (14.18). Then, there exists a multiplier $\lambda^* \in R$ such that the following conditions hold*

- (i) $Bs^* + g + \lambda^*s = 0$,
- (ii) $\|s^*\| \leq \Delta$,
- (iii) $\lambda^*(\|s^*\| - \Delta) = 0$,
- (iv) $\lambda^* \geq 0$.

Proof In order to prove the thesis, problem (14.18) is rewritten in the following equivalent form

$$\begin{aligned} \min_{s \in R^n} m(s) &= f + g^T s + \frac{1}{2} s^T B s \\ &\quad \|s\|^2 \leq \Delta^2, \end{aligned} \quad (14.24)$$

where the function defining the constraint is continuously differentiable.

Hence, we can refer to the optimality conditions of constrained problems with continuously differentiable functions.

Note that a *constraint qualification condition* is satisfied, i.e., the active constraint gradients are linearly independent. Indeed, the gradient of the constraint is $2s$, which is not equal to zero if the constraint is active, that is, if $\|s\| = \Delta$. Then, the Karush-Kuhn-Tucker conditions hold at a local minimizer s^* .

Consider the Lagrangian function

$$L(s, \lambda) = \frac{1}{2}s^T Bs + g^T s + f + \lambda (\|s\|^2 - \Delta^2).$$

We have that there exists a scalar $\hat{\lambda}$ such that

- (i) $\nabla_s L(s^*, \hat{\lambda}) = 0,$
- (ii) $\|s^*\|^2 - \Delta^2 \leq 0,$
- (iii) $\hat{\lambda} (\|s^*\|^2 - \Delta^2) = 0,$
- (iv) $\hat{\lambda} \geq 0.$

Taking into account that

$$\nabla_s L(s^*, \hat{\lambda}) = Bs^* + g + 2\hat{\lambda}s,$$

and setting $\lambda^* = 2\hat{\lambda}$ the thesis is proved. \square

We can state *necessary and sufficient* optimality conditions of *global minimum*.

Proposition 14.6 (Global Minimum Conditions) *The point s^* is a global minimum point of problem (14.18) if and only if there exists a multiplier $\lambda^* \in R$ such that the following conditions hold*

- (i) $Bs^* + g + \lambda^*s^* = 0,$
- (ii) $\|s^*\| \leq \Delta,$
- (iii) $\lambda^*(\|s^*\| - \Delta) = 0,$
- (iv) $\lambda^* \geq 0,$
- (v) *the matrix $B + \lambda^*I$ is positive semidefinite.*

Proof (Necessary Conditions) Assume that s^* is a global minimizer. Then, the Karush-Kuhn-Tucker conditions of Proposition 14.5 hold, and hence assertions (i)–(iv) are proved. In order to prove assertion (v), we preliminarily observe that if $s^* = 0$ then, from (i), we have $g = 0$. In this case, $\|s^*\| < \Delta$ and hence s^* satisfies the unconstrained optimality conditions for the quadratic function $m(s) = 1/2s^T Bs$, implying that the Hessian matrix B must be positive semidefinite, so that (v) is proved.

Therefore, we can assume $s^* \neq 0$. By contradiction, suppose that (v) does not hold. Then, there exists a vector $\hat{z} \in R^n$ such that

$$\hat{z}^T (B + \lambda^*I)\hat{z} < 0. \quad (14.25)$$

We can assume that there exists a vector $z \in R^n$ such that

$$z^T(B + \lambda^* I)z < 0, \quad \text{and} \quad z^T s^* \neq 0. \quad (14.26)$$

In fact, if $\hat{z}^T s^* \neq 0$, we can set $z = \hat{z}$. Otherwise, if $\hat{z}^T s^* = 0$, we can set $z = \hat{z} + \alpha s^*$. This point is such that $z^T s^* = \hat{z}^T s^* + \alpha \|s^*\|^2 \neq 0$, moreover, by continuity, there exists $\varepsilon > 0$ such that $z^T(B + \lambda^* I)z < 0$ for all $z \in R^n$ such that $\|z - \hat{z}\| = \alpha \|s^*\| < \varepsilon$.

Now we show that (14.26) leads to a contradiction with the assumption that s^* is a global minimizer, i.e., we show that we can define a feasible point s such that $m(s) < m(s^*)$. To this aim, consider the point $s = s^* + \eta z$, where z is such that (14.26) is satisfied. Define η in such a way that the point $s^* + \eta z$ is a feasible point. Indeed we have:

$$\|s\|^2 = \|s^* + \eta z\|^2 = (s^* + \eta z)^T(s^* + \eta z) = \|s^*\|^2 + \eta^2 \|z\|^2 + 2\eta z^T s^*.$$

Choosing $\eta \neq 0$ such that $\eta^2 \|z\|^2 + 2\eta z^T s^* = 0$, and hence, setting

$$\eta = -\frac{2z^T s^*}{\|z\|^2},$$

we obtain $\|s\|^2 = \|s^*\|^2 \leq \Delta^2$. We prove that the point $s^* + \eta z$ is such that $m(s^* + \eta z) < m(s^*)$. Taking into account that (i) implies $Bs^* + g = -\lambda^* s^*$, we can write:

$$\begin{aligned} m(s^* + \eta z) - m(s^*) &= \eta \nabla m(s^*)^T z + \frac{1}{2} \eta^2 z^T B z = \eta (Bs^* + g)^T z + \frac{1}{2} \eta^2 z^T B z \\ &= -\eta \lambda^* z^T s^* + \frac{1}{2} \eta^2 z^T B z - \frac{1}{2} \eta^2 \lambda^* \|z\|^2 + \frac{1}{2} \eta^2 \lambda^* \|z\|^2 \\ &= \frac{1}{2} \eta^2 z^T (B + \lambda^* I) z - \frac{1}{2} \eta^2 \lambda^* \|z\|^2 - \eta \lambda^* z^T s^* \\ &= \frac{1}{2} \eta^2 z^T (B + \lambda^* I) z. \end{aligned} \quad (14.27)$$

Taking into account (14.26), we have $m(s^* + \eta z) - m(s^*) < 0$, and this contradicts the assumption that s^* is a global minimizer, so that, (v) is necessarily satisfied.

Sufficient Conditions Assume that s^* and λ^* are such that (i)–(v) hold. From (i) and (v) it follows that the quadratic function $p(s) = \frac{1}{2} s^T (B + \lambda^* I) s + g^T s$ is a convex function having a stationary point at s^* . From known results it follows that s^* is an unconstrained global minimizer of $p(s)$ and hence we can write for all $s \in R^n$:

$$\frac{1}{2} s^T (B + \lambda^* I) s + g^T s \geq \frac{1}{2} s^{*T} (B + \lambda^* I) s^* + g^T s^*. \quad (14.28)$$

Then, for all $s \in R^n$ we have:

$$\frac{1}{2} s^T B s + g^T s \geq \frac{1}{2} s^{*T} B s^* + g^T s^* + \frac{\lambda^*}{2} (\|s^*\|^2 - \|s\|^2). \quad (14.29)$$

If $\|s^*\| < \Delta$ then, from condition (iii) it follows $\lambda^* = 0$, so that, (14.29) implies $m(s) \geq m(s^*)$, namely that s^* is a global minimizer on R^n for the objective function $m(s)$. As a consequence, the point s^* is a global minimizer for the constrained problem (14.18).

If $\|s^*\| = \Delta$ then, using condition (iv), we can write

$$\frac{\lambda^*}{2}(\|s^*\|^2 - \|s\|^2) = \frac{\lambda^*}{2}(\Delta^2 - \|s\|^2) \geq 0$$

for all s such that $\|s\| \leq \Delta$. From (14.29) it follows that s^* is a global minimizer for problem (14.18), and this completes the proof. \square

The optimality conditions stated by Proposition 14.6 lead to define a strategy for computing the exact solution s^* of problem (14.18).

The basic idea can be summarized as follows:

- I check if $\lambda^* = 0$ is such that conditions (i) and (v) of Proposition 14.6 hold, with $\|s^*\| \leq \Delta$;
- II otherwise, determine a sufficiently high value of λ in such a way that the matrix $B + \lambda I$ is positive semidefinite and we have

$$\|s(\lambda)\| = \Delta, \quad (14.30)$$

where

$$(B + \lambda I)s(\lambda) = -g.$$

Note that (14.30) consists in the problem of determining a solution of a nonlinear equation in one variable.

14.7 Trust Region Approach to Globalizing Newton's Method

In this section we show that a trust region algorithm, based on the quadratic model whose matrix B_k is the Hessian matrix $\nabla^2 f(x_k)$, is a *globally convergent modification* of Newton's method according to the definition given in Chap. 13.

Consider the algorithm defined in Sect. 14.1 and assume that the quadratic model (14.2) takes the form

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s_k^T \nabla^2 f(x_k) s, \quad (14.31)$$

so that, the subproblem considered at Step 2 is the following

$$\begin{aligned} \min m_k(s) &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s \\ \|s\| &\leq \Delta_k. \end{aligned} \quad (14.32)$$

Now we present a conceptual scheme, called *Newton Trust Region* (NTR), and we prove that it is a globally convergent modification of Newton's method.

At any iteration k :

I. Compute (if possible) the Newton's direction

$$s^N(x_k) = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k).$$

II. If $\|s^N(x_k)\| \leq \Delta_k$ and $s^N(x_k)$ satisfies the condition

$$m_k(0) - m_k(s^N(x_k)) \geq c_1 \|\nabla f(x_k)\| \min \left(\Delta_k, \frac{\|\nabla f(x_k)\|}{1 + \|\nabla^2 f(x_k)\|} \right), \quad (14.33)$$

with $c_1 \in (0, 1/2)$; if (14.33) holds then set $s_k = s^N(x_k)$, otherwise determine a direction s_k such that $\|s_k\| \leq \Delta_k$ and

$$m_k(0) - m_k(s_k) \geq c_1 \|\nabla f(x_k)\| \min \left(\Delta_k, \frac{\|\nabla f(x_k)\|}{1 + \|\nabla^2 f(x_k)\|} \right). \quad (14.34)$$

Concerning the direction $s_k \neq s_k^N$ of point (II), it is possible to employ, for instance, the Cauchy step, or the exact solution of the trust region subproblem, or an approximate solution computed by either the dogleg method or the conjugate gradient method. Indeed, in all these cases, condition (14.34) holds.

We can prove that NTR is a *globally convergent Newton-type algorithm*, i.e., denoting by $\{x_k\}$ the generated sequence, we have

- (i) there exist limit points of $\{x_k\}$ and every limit point is a stationary point of f belonging to the compact level set \mathcal{L}_0 ;
- (ii) if $\{x_k\}_K$ is an infinite subsequence converging to a limit point, there exists at least an infinite subsequence of it such that the objective function is strictly decreasing;
- (iii) if $\{x_k\}$ converges towards a local minimum point x^* of f and $\nabla^2 f$ is a positive definite matrix that satisfies the assumptions of Proposition 13.2, then there exists k^* such that, for all $k \geq k^*$ we have

$$s_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

and hence the iteration of the algorithm is the pure Newton iteration.

Preliminarily we state the following result.

Lemma 14.2 *Assume that*

$$\lambda_m(\nabla^2 f(x_k)) \geq \epsilon > 0, \quad (14.35)$$

where $\lambda_m(\nabla^2 f(x_k))$ is the minimum eigenvalue of $\nabla^2 f(x_k)$ and that $m_k(s_k) < m_k(0)$. Then we have

$$\|s_k\| \leq \frac{2}{\epsilon} \|\nabla f(x_k)\|. \quad (14.36)$$

Proof Let

$$\phi(t) = m_k(0) - m_k(ts_k) = -t \nabla f(x_k)^T s_k - \frac{1}{2} t^2 s_k^T \nabla^2 f(x_k) s_k,$$

for $t > 0$. Function $\phi(t)$ is a concave quadratic function (as consequence of hypothesis (14.35)) such that $\phi(0) = 0$, and $\phi(1) > 0$ since we are assuming $m_k(s_k) < m_k(0)$. Therefore we have

$$t^* = \arg \max_t \phi(t) > 1/2,$$

being $\phi(t)$ symmetric with respect to t^* . Using the Cauchy-Schwarz inequality and (14.35), we can write

$$\frac{1}{2} < t^* = \frac{|\nabla f(x_k)^T s_k|}{s_k^T \nabla^2 f(x_k) s_k} \leq \frac{\|\nabla f(x_k)\| \|s_k\|}{\|s_k\|^2 \epsilon} = \frac{\|\nabla f(x_k)\|}{\|s_k\| \epsilon},$$

so that, the thesis is proved. \square

Proposition 14.7 *Let $f : R^n \rightarrow R$ be a twice continuously differentiable function over R^n . Assume that the level set \mathcal{L}_0 is compact and that the gradient is Lipschitz-continuous over \mathcal{L}_0 . Then Algorithm NTR is a globally convergent Newton-type algorithm in the sense of Definition 13.1.*

Proof The instructions of the algorithm imply that $f(x_{k+1}) \leq f(x_k)$ and hence that the points of the sequence $\{x_k\}$ belong to the compact level set \mathcal{L}_0 . Thus $\{x_k\}$ admits limit points and every limit point belongs to \mathcal{L}_0 . The continuity of $\nabla^2 f$ and the

compactness of \mathcal{L}_0 imply that there exists a number $\beta > 0$ such that $\|\nabla^2 f(x_k)\| \leq \beta$ for all $k \geq 0$. Furthermore we have that the direction s_k satisfies the condition

$$m_k(0) - m_k(s_k) \geq c_1 \|\nabla f(x_k)\| \min\left(\Delta_k, \frac{\|\nabla f(x_k)\|}{1 + \|\nabla^2 f(x_k)\|}\right), \quad (14.37)$$

with $c_1 \in (0, 1/2)$. Then, condition (i) of Definition 13.1 follows from (14.37) and Proposition 14.2.

We are assuming that the algorithm does not terminate. Let $\{x_k\}_K$ be any infinite sequence convergent to a stationary point. Then there exists a further infinite subsequence of $\{x_k\}_K$ made of distinct points. Then, condition (ii) of Definition 13.1 holds recalling that, according to the instructions of the trust region algorithm, we have $x_{k+1} \neq x_k$ if and only if $f(x_{k+1}) < f(x_k)$.

In order to prove condition (iii), suppose that there exists a limit point x^* such that $\nabla^2 f(x^*)$ is positive definite. We are assuming that $\nabla^2 f$ is Lipschitz-continuous in a neighborhood of x^* . Let $\{x_k\}_K$ be the subsequence converging to x^* .

First we show that the whole sequence $\{x_k\}$ converges to x^* . Taking into account the continuity of $\nabla^2 f$ we have that there exists a neighborhood of x^* with radius r , denoted by $B(x^*; r)$, where $\nabla^2 f(x)$ is positive definite. We also have $\nabla f(x^*) = 0$. Then, the direction

$$s^N(x_k) = \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

tends to zero for $k \in K$ and $k \rightarrow \infty$. Therefore, there exists a radius $r_1 \leq r/4$ such that, for $k \in K$ and k sufficiently large, we have

$$x_k \in B(x^*; r_1) \quad \|s^N(x_k)\| < r/2. \quad (14.38)$$

We denote by \hat{f} the minimum value of $f(x)$, with $x \in B(x^*; r) \setminus B(x^*; r_1)$. The function f is a strictly convex function over $B(x^*; r)$ and x^* is its global minimum. Since the sequence $\{f(x_k)\}$ monotonically converges to $f(x^*)$, we can find an index $k_1 \in K$ such that $x_{k_1} \in B(x^*; r_1)$ and $f(x_{k_1}) < \hat{f}$.

Now we show that the whole sequence $\{x_k\}$ is contained in $B(x^*; r_1)$. If this is not true then there must exist a point $x_{\hat{k}} \in B(x^*; r_1)$ with $\hat{k} \geq k_1$ such that the successive point $x_{\hat{k}+1}$ does not belong to $B(x^*; r_1)$. However, the point $x_{\hat{k}+1}$ can not belong to $B(x^*; r) \setminus B(x^*; r_1)$ (otherwise, as $\{f(x_k)\}$ is a decreasing sequence, we would obtain a contradiction with $f(x_{\hat{k}}) \leq f(x_{k_1}) < \hat{f}$) and hence it must be outside $B(x^*; r)$. This implies $\|s_{\hat{k}}\| > 3r/4$ and hence $\Delta_{\hat{k}} \geq 3r/4$. From (14.38) it follows $\|s^N(x_{\hat{k}})\| \leq \Delta_{\hat{k}}$, so that, using Proposition 14.6 we obtain $s_{\hat{k}} = s^N(x_{\hat{k}})$. Recalling again (14.38) we have $\|s_{\hat{k}}\| = \|s^N(x_{\hat{k}})\| < r/2$ and this contradicts $\|s_{\hat{k}}\| > 3r/4$.

Then we can state that the whole sequence remains in $B(x^*; r_1)$ with r_1 arbitrarily small. On the other hand, in this neighborhood there can not exist distinct limit points. Indeed, the strict convexity of f over $B(x^*; r_1)$ implies that there can not

exist a stationary point different from x^* . Therefore, we can conclude that the whole sequence converges to x^* .

Taking into account the continuity of $\nabla^2 f$, for k sufficiently large we have that there exist two constants $\epsilon, M > 0$ such that

$$\lambda_m \left(\nabla^2 f(x_k) \right) \geq \epsilon > 0 \quad \| \nabla^2 f(x_k) \| \leq M.$$

Using condition (14.37) and Lemma 14.2 we can write

$$\begin{aligned} m_k(0) - m_k(s_k) &\geq c_1 \frac{2}{\epsilon} \|s_k\| \min \left(\Delta_k, \frac{2\|s_k\|}{\epsilon(1+M)} \right) \\ &\geq c_1 \frac{2}{\epsilon} \|s_k\| \min \left(\|s_k\|, \frac{2\|s_k\|}{\epsilon(1+M)} \right) \\ &= \frac{1}{2} m \|s_k\|^2, \end{aligned}$$

where

$$\frac{1}{2} m = c_1 \frac{2}{\epsilon} \min \left(1, \frac{2}{\epsilon(1+M)} \right)$$

We also have

$$|\rho_k - 1| = \left| \frac{f(x_k) - f(x_k + s_k) - (m_k(0) - m_k(s_k))}{m_k(0) - m_k(s_k)} \right| = \left| \frac{m_k(s_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)} \right|.$$

From the assumptions stated we get

$$f(x_k + s_k) = f(x_k) + \nabla f(x_k)^T s_k + \frac{1}{2} s_k^T \nabla^2 f(\xi_k) s_k,$$

where $\xi_k = x_k + \theta_k s_k$, with $\theta_k \in (0, 1)$. Using the above condition we get

$$|\rho_k - 1| = \left| \frac{1/2 s_k^T [\nabla^2 f(x_k) - \nabla^2 f(\xi_k)] s_k}{m_k(0) - m_k(s_k)} \right| \leq \frac{1/2 \|s_k\|^2 L_H \|s_k\|}{1/2 m \|s_k\|^2} = \frac{L_H}{m} \|s_k\|,$$

where $L_H > 0$ is the Lipschitz constant relative to the Hessian matrix.

As $s_k \rightarrow 0$ (this follows from the fact that $\{x_k\}$ converges to x^* and from Lemma 14.2) we will have for k sufficiently large $\rho_k > \eta_2$. Then the instructions of the trust region strategy imply that for k sufficiently large the radius Δ_k is never reduced, and hence we have

$$\Delta_k \geq \bar{\Delta} > 0.$$

For k sufficiently large the Newton direction $s^N(x_k)$ is well-defined. Furthermore, as $\Delta_k \geq \bar{\Delta}$ and $s^N(x_k) \rightarrow 0$, we have $\|s^N(x_k)\| < \Delta_k$, so that, from Proposition 14.6 it follows that $s^N(x_k)$ is the solution of the strictly convex problem (14.32). Therefore, for k sufficiently large condition (14.33) holds. This implies that $s_k = s^N(x_k)$, and this concludes the proof. \square

14.8 Complexity Issues and Adaptive Regularized Methods

We have seen in Chap. 11 that the upper bound $O(\epsilon^{-2})$ on the number of iterations required to obtain an approximate stationary point was given by several variants of the steepest descent algorithm.

Concerning the pure Newton's method for twice continuously differentiable functions, it has been constructed [39] a two-dimensional example on which the algorithm requires $O(\epsilon^{-2})$ iterations to obtain an approximate stationary point, i.e., converges at a rate which corresponds to the worst-case known for the steepest descent method on general nonconvex objective functions. Then, Newton's method may be as slow as steepest descent in the worst case.

Adaptive regularized methods have been recently studied as an alternative to classical globalization techniques of Newton's method. The Adaptive Regularization with Cubic (ARC) of Newton's method gives rise to a local cubic overestimator of the objective function f which is employed for computing the step from one iterate to the next one. Under mild assumptions, ARC iterates converge to first-order critical points.

The distinguishing features of ARC from linesearch and trust-region techniques are the results on worst-case iteration and gradient evaluation complexity. Specifically, a worst-case iteration count of order $O(\epsilon^{-3/2})$ has been established to drive the norm of the gradient of f below a prefixed accuracy ϵ . This bound holds for the class of unconstrained optimization problems where:

- the objective function f is twice continuously differentiable;
- the Hessian matrix $\nabla^2 f$ is Lipschitz continuous, that is, there exists a positive constant L such that, for all $x, y \in R^n$,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|.$$

and assuming that the objective function is bounded below. The good complexity bound of ARC can be achieved by minimizing the cubic model approximately within some suitable accuracy requirement. In order to introduce the algorithm

presented in [22], let us consider the idea leading to the cubic model. Under the assumptions stated above, using the Taylor's theorem, we can write

$$\begin{aligned}
 f(x_k + s) &= f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \\
 &\int_0^1 (1-t) s^T \left[\nabla^2 f(x_k + ts) - \nabla^2 f(x_k) \right] s dt \\
 &\leq f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{L}{3} \|s\|^3 = m_k^C(s).
 \end{aligned} \tag{14.39}$$

Thus, for every step s such that $m_k^C(s) \leq m_k^C(0) = f(x_k)$, the point $x_k + s$ improves f . In order to define a model of practical interest, the constant L may be replaced by a dynamic positive parameter σ_k . This gives rise to the model

$$m_k(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{\sigma_k}{3} \|s\|^3.$$

The rules for updating the parameter σ_k take into account the agreement between f and m_k and parallel those for updating the radius in trust-region methods. In particular, σ_k can be viewed as the reciprocal of the trust-region radius.

Before to describe the algorithm, we define

$$m(x, s, \sigma) = T_2(x, s) + \frac{\sigma}{3} \|s\|^3, \tag{14.40}$$

whose gradient is

$$\nabla_s m(x, s, \sigma) = \nabla_s T_2(x, s) + \sigma \|s\| s. \tag{14.41}$$

Furthermore, we have

$$m(x, 0, \sigma) = T_2(x, 0) = f(x).$$

The formal scheme of an ARC method is reported below.

Conceptual Model of an ARC Method

Data: Starting point $x_0 \in R^n$, $\sigma_0 > 0$; positive constants $\eta_1, \eta_2, \gamma_1, \gamma_2, \gamma_3$ such that

$$\theta > 0 \quad \sigma_{min} \in (0, \sigma_0] \quad 0 < \eta_1 \leq \eta_2 < 1 \quad 0 < \gamma_1 < 1 < \gamma_2 < \gamma_3 \tag{14.42}$$

Set $k = 0$.

(continued)

Step1: Step calculation. Compute the step s_k by approximately minimizing the model $m(x_k, s_k, \sigma_k)$ w.r.t. s , in the sense that conditions

$$m(x_k, s_k, \sigma_k) < m(x_k, 0, \sigma_k), \quad (14.43)$$

and

$$\|\nabla_s m(x_k, s_k, \sigma_k)\| \leq \theta \|s_k\|^2 \quad (14.44)$$

hold.

Step 2: Acceptance of the trial point. Compute $f(x_k + s)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{T_2(x_k, 0) - T_2(x_k, s_k)}. \quad (14.45)$$

If $\rho_k \geq \eta_1$ then set $x_{k+1} = x_k + s_k$, otherwise set $x_{k+1} = x_k$.

Step 3: Regularization parameter update. Define σ_{k+1} as follows

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_1 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (14.46)$$

Set $k = k + 1$ and go to Step 1.

We observe that at Step 1 it is required an approximate minimization of $m(x_k, s, \sigma_k)$, i.e., to find a step s_k yielding a decrease of the cubic model (see condition (14.43)) and approximating a stationary point (see condition (14.44)).

The rule for updating the regularization parameter follows the same strategy for updating the radius in a trust region framework. The updating depends on the agreement between the actual reduction $f(x_k) - f(x_k + s_k)$ and the predicted reduction $f(x_k) - T_2(x_k, s_k)$ according to the quadratic model.

As proved in [22], the complexity bound for ARC is $O(\epsilon^{-3/2})$.

14.9 Appendix: Proofs of Convergence

In this appendix we prove Propositions 14.1 and 14.2. Preliminarily we state the following lemma.

Lemma 14.3 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a continuously differentiable function over \mathbb{R}^n and suppose that there exists a constant $\beta > 0$ such that $\|B_k\| \leq \beta$ for every k . Assume that for k sufficiently large we have

$$m_k(0) - m_k(s_k) \geq a \min \{\Delta_k, b\}, \quad (14.47)$$

where a, b are positive numbers. Then there exists a value $\bar{\Delta} > 0$ such that for k sufficiently large we have

$$\Delta_k \geq \bar{\Delta}. \quad (14.48)$$

Proof First we state some relations used later. Using the definition (14.5) of ρ_k we can write

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k) - f(x_k + s_k) - (m_k(0) - m_k(s_k))}{m_k(0) - m_k(s_k)} \right| \\ &= \left| \frac{m_k(s_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)} \right|. \end{aligned} \quad (14.49)$$

From the mean value theorem we have

$$f(x_k + s_k) = f(x_k) + \nabla f(x_k)^T s_k + [\nabla f(x_k + t_k s_k) - \nabla f(x_k)]^T s_k,$$

with $t_k \in (0, 1)$, from which it follows

$$\begin{aligned} |m_k(s_k) - f(x_k + s_k)| &= \left| \frac{1}{2} s_k^T B_k s_k - [\nabla f(x_k + t_k s_k) - \nabla f(x_k)]^T s_k \right| \\ &\leq \frac{\beta}{2} \|s_k\|^2 + h(s_k) \|s_k\|, \end{aligned} \quad (14.50)$$

where $h(s_k)$ is such that $h(s_k) \rightarrow 0$ if $s_k \rightarrow 0$.

Using (14.49), (14.50), (14.47), and taking into account the trust region constraint $\|s_k\| \leq \Delta_k$ we obtain

$$|\rho_k - 1| \leq \frac{\Delta_k (\beta \Delta_k / 2 + h(s_k))}{a \min \{\Delta_k, b\}}. \quad (14.51)$$

In order to prove the thesis, assume by contradiction that (14.48) does not hold. Let $K \subseteq \{0, 1, \dots\}$ the infinite subset such that

$$\Delta_{k+1} < \Delta_k \quad \forall k \in K \quad \Delta_{k+1} \geq \Delta_k \quad \forall k \notin K. \quad (14.52)$$

Note that if (14.48) is not true then the infinite subset K exists. We can define a further subset of K (called again K) such that

$$\lim_{k \in K, k \rightarrow \infty} \Delta_{k+1} = \lim_{k \in K, k \rightarrow \infty} \Delta_k = 0, \quad (14.53)$$

where we have used the condition $\Delta_{k+1} \geq \gamma_1 \Delta_k$ for $k \in K$.

For $k \in K$ and k sufficiently large we have

$$\Delta_k \leq b \quad \beta \Delta_k / 2 + h(s_k) \leq a(1 - \eta_2),$$

so that, from (14.51) it follows

$$|\rho_k - 1| \leq \frac{\Delta_k a(1 - \eta_2)}{a \Delta_k} = 1 - \eta_2.$$

Therefore we must have $\rho_k \geq \eta_2$ for $k \in K$ and k sufficiently large. The instructions of the algorithm imply $\Delta_{k+1} \geq \Delta_k$ for $k \in K$ and k sufficiently large, and this contradicts (14.52). \square

Proposition 14.8 *Let $f : R^n \rightarrow R$ be a continuously differentiable function over R^n . Suppose that f is bounded below and that there exists a positive constant β such that $\|B_k\| \leq \beta$ for every k . Then*

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (14.54)$$

Proof The instructions of the algorithm imply $f(x_{k+1}) \leq f(x_k)$ for every k , so that, taking into account that f is bounded below, we obtain

$$\lim_{k \rightarrow \infty} f(x_k) - f(x_{k+1}) = 0. \quad (14.55)$$

In order to prove the thesis, assume by contradiction that there exist $\epsilon > 0$ and an index \bar{k} such that for every $k \geq \bar{k}$ we have

$$\|\nabla f(x_k)\| \geq \epsilon. \quad (14.56)$$

From (14.7) it follows for $k \geq \bar{k}$

$$m_k(0) - m_k(s_k) \geq c_1 \epsilon \min \left\{ \Delta_k, \frac{\epsilon}{1 + \beta} \right\}. \quad (14.57)$$

Lemma 14.3 implies that there exists a scalar $\tilde{\Delta} > 0$ such that

$$\Delta_k \geq \tilde{\Delta} \quad (14.58)$$

for k sufficiently large.

Suppose that there exists an infinite subset \tilde{K} such that

$$\rho_k \geq \eta_1 \quad \forall k \in \tilde{K}. \quad (14.59)$$

From (14.57) it follows for $k \in \tilde{K}$ and $k \geq \bar{k}$

$$\begin{aligned} f(x_k) - f(x_{k+1}) &= f(x_k) - f(x_k + s_k) \\ &\geq \eta_1 (m_k(0) - m_k(s_k)) \\ &\geq \eta_1 c_1 \epsilon \min \{ \Delta_k, \epsilon / (1 + \beta) \}. \end{aligned} \quad (14.60)$$

Using (14.55) and (14.60) we obtain

$$\lim_{k \in \tilde{K}, k \rightarrow \infty} \Delta_k = 0,$$

and this contradicts (14.58). Therefore, there can not exist an infinite subset \tilde{K} for which (14.59) holds. Then for k sufficiently large we have $\rho_k < \eta_1$, from which, taking into account the instructions of the algorithm, we get

$$\Delta_{k+1} \leq \gamma_2 \Delta_k,$$

with $0 < \gamma_2 < 1$. The above inequality implies $\Delta_k \rightarrow 0$ for $k \rightarrow \infty$ and this contradicts (14.58). \square

Proposition 14.9 *Let $f : R^n \rightarrow R$ be a continuously differentiable function over R^n . Suppose that f is bounded below and that there exists a positive constant β such that $\|B_k\| \leq \beta$ for every k . Assume that the gradient is*

(continued)

Proposition 14.9 (continued)

Lipschitz-continuous over the level set \mathcal{L}_0 , i.e., there exists a constant $L > 0$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathcal{L}_0. \quad (14.61)$$

Then

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0. \quad (14.62)$$

Proof The instructions of the algorithm imply $f(x_{k+1}) \leq f(x_k)$ and hence the points of the generated sequence $\{x_k\}$ belong to the level set \mathcal{L}_0 . Furthermore, as f is bounded below, it follows

$$\lim_{k \rightarrow \infty} f(x_k) - f(x_{k+1}) = 0. \quad (14.63)$$

Let K_S be the set of *successful* iterates, that is, the iterates such that

$$x_{k+1} \neq x_k.$$

In order to prove (14.62) assume by contradiction that there exists an infinite subset $\{k_i\} \subset \{0, 1, \dots\}$ such that for every i we have

$$\|\nabla f(x_{k_i})\| \geq 2\epsilon > 0. \quad (14.64)$$

Note that, relabelling if necessary the sequence $\{k_i\}$, we can assume $\{k_i\} \subseteq K_S$ since for every $k_i \notin K_S$ we have $x_{k_i} = x_{k_i-1}$ and hence, by induction, it follows

$$x_{k_i} = x_{k_i-1} = \dots = x_{k_i-h_i},$$

where $k_i - h_i \in K_S$.

For every i let $l(k_i) > k_i$ be the smallest integer such that

$$\|\nabla f(x_{l(k_i)})\| \leq \epsilon. \quad (14.65)$$

Note that Proposition 14.1 ensures that the index $l(k_i)$ is well-defined (to simplify the notation we set $l_i = l(k_i)$). Using similar reasonings explained above we can assume that the sequence $\{l_i\}$ is contained in K_S .

Then we can write

$$\|\nabla f(x_k)\| > \epsilon \quad k_i \leq k < l_i \quad \|\nabla f(x_{l_i})\| \leq \epsilon. \quad (14.66)$$

Now consider the subset of successful iterates

$$K = \{k \in K_S : k_i \leq k < l_i\}.$$

Using (14.7) and (14.66), for every $k \in K$ we can write

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 [m_k(0) - m_k(x_k + s_k)] \\ &\geq c_1 \epsilon \eta_1 \min \{\Delta_k, \epsilon/(1+\beta)\}. \end{aligned} \tag{14.67}$$

From (14.63) and (14.67) we obtain

$$\lim_{k \in K, k \rightarrow \infty} \Delta_k = 0. \tag{14.68}$$

Thus, using again (14.67), for $k \in K$ and k sufficiently large we have

$$\Delta_k \leq \frac{1}{c_1 \epsilon \eta_1} [f(x_k) - f(x_{k+1})]. \tag{14.69}$$

For i sufficiently large we can write

$$\begin{aligned} \|x_{k_i} - x_{l_i}\| &\leq \sum_{j=k_i, j \in K}^{l_i-1} \|x_j - x_{j+1}\| \leq \sum_{j=k_i, j \in K}^{l_i-1} \Delta_j \\ &\leq \frac{1}{c_1 \epsilon \eta_1} [f(x_{k_i}) - f(x_{l_i})], \end{aligned} \tag{14.70}$$

where the last inequality is obtained using (14.69). From (14.63) and (14.70) we get $\|x_{k_i} - x_{l_i}\| \rightarrow 0$ per $i \rightarrow \infty$. Assumption (14.9) implies that $\|\nabla f(x_{k_i}) - \nabla f(x_{l_i})\| \rightarrow 0$ for $i \rightarrow \infty$, and this contradicts the fact that, from (14.64) and (14.65), we have $\|\nabla f(x_{k_i}) - \nabla f(x_{l_i})\| \geq \epsilon$. \square

14.10 Exercises

14.1 Prove Proposition 14.3.

14.2 Define a computer code of the trust region algorithm using the Hessian matrix $\nabla^2 f(x_k)$ as B_k and the *dogleg* method for determining an approximate solution of the subproblem.

14.11 Notes and References

Primary reference books for trust region methods are [50] and [196]. The matter of this chapter is mainly based on the above books. The dogleg method was introduced in [213], and the conjugate gradient method for the trust region subproblem was presented in [245]. Adaptive regularized methods, briefly analyzed in the chapter, have been recently studied as an alternative to classical globalization techniques for nonlinear constrained and unconstrained optimization. The distinguishing features of adaptive regularized methods from linesearch and trust-region techniques are the results on worst-case iteration and gradient evaluation complexity. The use of a cubic overestimator of the objective function was first considered in [120] and later exploited in several works. The literature on adaptive regularized methods is wide and constantly evolving, we limit ourselves to cite two seminal papers, [40] and [41].

Chapter 15

Quasi-Newton Methods



In this chapter we give a short introduction to *Quasi-Newton methods* (also known as *variable metric methods* or *secant methods*), which constitute an important class of unconstrained minimization methods that use only first order derivatives.

15.1 Preliminaries

Quasi-Newton methods yield an “approximation” to Newton’s method, which is not based, in general, on a consistent approximation of the Hessian matrix, but yet possesses, under appropriate assumptions, a superlinear convergence rate.

We already know that Newton’s method for unconstrained minimization is described by the iteration

$$x_{k+1} = x_k - \left[\nabla^2 f(x_k) \right]^{-1} \nabla f(x_k).$$

Quasi-Newton methods are defined by an iteration of a similar structure:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k),$$

where now B_k is a matrix, updated at each step, which approximates (in a sense to be specified) the Hessian matrix and α_k is the step-size along the search direction.

When the objective function is quadratic, that is

$$f(x) = \frac{1}{2} x^T Q x - c^T x,$$

where Q is a symmetric matrix, we have that $\nabla f(x) = Qx - c$ and hence, given two points x and y in R^n , we can write

$$\nabla f(y) - \nabla f(x) = Q(y - x).$$

If Q is nonsingular we can also write

$$Q^{-1}(\nabla f(y) - \nabla f(x)) = y - x.$$

Thus, in the general case, we can think of determining the matrix B_{k+1} in a way that the following condition (known as *Quasi-Newton equation*) is satisfied.

$$\nabla f(x_{k+1}) - \nabla f(x_k) = B_{k+1}(x_{k+1} - x_k). \quad (15.1)$$

This implies that we impose a condition that the Hessian matrix actually satisfies when f is quadratic. Letting:

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k), \end{aligned}$$

we can determine B_{k+1} by updating B_k in a way that

$$B_{k+1} = B_k + \Delta B_k, \quad \text{with } y_k = (B_k + \Delta B_k) s_k. \quad (15.2)$$

Following a similar reasoning we can refer to the Quasi-Newton equation written in the form

$$H_{k+1} (\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k, \quad (15.3)$$

where now H_{k+1} is intended as an approximation to the inverse of the Hessian. In this case, a Quasi-Newton iteration is of the form

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

and the updating rule becomes:

$$H_{k+1} = H_k + \Delta H_k, \quad \text{with } (H_k + \Delta H_k) y_k = s_k. \quad (15.4)$$

The various Quasi-Newton methods proposed up to now differs essentially in the correction terms ΔB_k or ΔH_k that appear in the updating formulas (15.2) or (15.4). We call *direct formulae* the updating formulae for B_k and *inverse formulae* those for H_k .

The Quasi-Newton method can be also referred to a system of nonlinear equations $F(x) = 0$, where $F : R^n \rightarrow R^n$. In this case the Quasi-Newton methods

can be viewed as n -dimensional extensions of the secant method and can be defined through the direct formula

$$x_{k+1} = x_k - \alpha_k B_k^{-1} F(x_k), \quad (15.5)$$

or the inverse formula

$$x_{k+1} = x_k - \alpha_k H_k F(x_k), \quad (15.6)$$

where B_k and H_k can be viewed as approximations, respectively, of $J(x_k)$ or $J(x_k)^{-1}$, and $J(x)$ is the Jacobian matrix of F , assumed to be non singular.

Another interpretation of many Quasi-Newton methods can be that of viewing the search direction employed in the method as the steepest descent direction with respect to the non-Euclidean metric derived from the norm

$$\|x\|_{B_k} = [x^T B_k x]^{1/2}, \quad (15.7)$$

where B_k is assumed to be a positive definite symmetric matrix. In fact the direction that minimizes the directional derivative among those with $\|d\|_{B_k} = 1$ is given by

$$d_k = -\frac{B_k^{-1} \nabla f(x_k)}{[\nabla f(x_k)^T B_k^{-1} \nabla f(x_k)]^{1/2}}.$$

As B_k varies with k , Quasi-Newton methods have been often defined as *variable metric methods*.

In the sequel we will give a short description of two class of Quasi-Newton methods corresponding to *updating formulae of rank one*, typically used in the solution of systems of nonlinear equations, and *updating formulae of rank two*, which are preferred in the context of optimization problems.

We recall here a formula for the computation of the inverse of a matrix updated with the addition of a term of special structure, known as *Sherman-Morrison-Woodbury formula*. See, e.g. [200, p. 50].

Proposition 15.1 (Sherman-Morrison-Woodbury Formula) *Let A be a nonsingular $n \times n$ matrix and let U, V be two $n \times m$ matrices with $m \leq n$.*

Then the matrix $A + UV^T$ is non singular if and only if the matrix

$$I + V^T A^{-1} U$$

(continued)

Proposition 15.1 (continued)
is nonsingular and, in this case, we have:

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U \left(I + V^T A^{-1}U \right)^{-1} V^T A^{-1}. \quad (15.8)$$

In particular, if $m = 1$ and $u, v \in R^n$ we have that the matrix $A + uv^T$ is nonsingular if and only if

$$1 + v^T A^{-1}u \neq 0$$

and, in this case, we have:

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}. \quad (15.9)$$

□

15.2 Rank 1 Formulae

Let us consider the Quasi-Newton equation in the form

$$y_k = (B_k + \Delta B_k) s_k. \quad (15.10)$$

This equation does not define uniquely the correction term ΔB_k and there exist infinitely many updating formulae that satisfy (15.10). One of the simplest formulae can be that of the form:

$$\Delta B_k = \rho_k u_k v_k^T, \quad (15.11)$$

where $\rho_k \in R$ and $u_k, v_k \in R^n$. The matrix $u_k v_k^T$ is a $n \times n$ matrix of rank 1 and hence the formulae based on (15.11) are called *rank 1 formulae*. By imposing condition (15.10) we obtain

$$y_k = B_k s_k + \rho_k u_k v_k^T s_k,$$

that can be satisfied by assuming

$$u_k = y_k - B_k s_k$$

$$\rho_k = 1/v_k^T s_k,$$

where v_k is an arbitrary vector that satisfies $v_k^T s_k \neq 0$. Then we have

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)v_k^T}{v_k^T s_k}, \quad (15.12)$$

In particular, assuming $v_k = s_k$ we obtain the updating formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k}, \quad (15.13)$$

known as *Broyden's formula*[36]. From (15.13), using Proposition 15.1 we can obtain from (15.9) the inverse formula

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) s_k^T H_k}{s_k^T H_k y_k}, \quad (15.14)$$

provided that $s_k^T H_k y_k \neq 0$.

Broyden's method is typically employed in the solution of nonlinear equations, but it is not used in unconstrained optimization. We can note, in particular that (15.13) does not guarantee that B_{k+1} is a symmetric matrix or that $-B_k^{-1} \nabla f(x_k)$ is a descent direction. It is possible to construct a symmetric rank 1 correction, by choosing, for instance in (15.12) $v_k = y_k - B_k s_k$, but in unconstrained optimization rank 2 methods are definitely preferred.

Globalization techniques for Broyden's method will be further considered in connection to methods for nonlinear equations.

15.3 Rank Two Formulae and the BFGS Method

Rank 2 formulae are those where the correction term ΔB_k or ΔH_k can be expressed as the sum of two dyads, which form a rank 2 matrix. In particular, if we refer to inverse formulae we can define a rank 2 updating formula by letting

$$H_{k+1} = H_k + a_k u_k u_k^T + b_k v_k v_k^T,$$

where $a_k, b_k \in R$ and $u_k, v_k \in R^n$. By imposing the Quasi-Newton condition

$$H_{k+1} y_k = s_k$$

we must have

$$H_k y_k + a_k u_k u_k^T y_k + b_k v_k v_k^T y_k = s_k. \quad (15.15)$$

One of the first Quasi-Newton methods proposed that satisfies (15.15) is the so called *Davidon-Fletcher-Powell* (DFP) method, where (15.15) is satisfied by assuming

$$u_k = s_k, \quad v_k = H_k y_k$$

$$a_k = 1/s_k^T y_k \quad b_k = -1/y_k^T H_k y_k.$$

This yields the DFP inverse formula

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}. \quad (15.16)$$

The next proposition shows under what conditions the matrix H_k remains positive definite if we start from a positive definite matrix H_0 .

Proposition 15.2 *Let H_k be a positive definite matrix. Then the matrix H_{k+1} given by (15.16) is positive definite if and only if*

$$s_k^T y_k > 0. \quad (15.17)$$

Proof (Sufficiency) By assumption we have $z^T H_k z > 0$ for all $z \in R^n, z \neq 0$. We show that

$$z^T H_{k+1} z > 0, \quad z \neq 0.$$

As H_k is positive definite we can use Cholesky factorization so that

$$H_k = LL^T,$$

where L is positive definite. Given $z \in R^n$ with $z \neq 0$ we can define the numbers

$$p = L^T z \quad q = L^T y_k. \quad (15.18)$$

Then we can write

$$\begin{aligned} z^T H_{k+1} z &= z^T H_k z + \frac{(s_k^T z)^2}{s_k^T y_k} - \frac{z^T L L^T y_k y_k^T L L^T z}{y_k^T L L^T y_k} \\ &= p^T p + \frac{(s_k^T z)^2}{s_k^T y_k} - \frac{(p^T q)^2}{q^T q} \\ &= \frac{\|p\|^2 \|q\|^2 - (p^T q)^2}{\|q\|^2} + \frac{(s_k^T z)^2}{s_k^T y_k} \end{aligned}.$$

By Schwarz inequality we have $\|p\|^2\|q\|^2 \geq (p^T q)^2$ and hence, as $s_k^T y_k > 0$ by assumption, we have $z^T H_{k+1} z \geq 0$. Moreover, as $z \neq 0$, if $z^T H_{k+1} z = 0$ we have

$$\|p\|^2\|q\|^2 - (p^T q)^2 = 0.$$

In this case, by Schwarz inequality, we must have $p = \lambda q$ for some $\lambda \in R$. As L^T is non singular, it follows from (15.18) that $z = \lambda y_k$, with $\lambda \neq 0$ (since we assumed $z \neq 0$). Then we have:

$$z^T H_{k+1} z = \lambda^2 \frac{(s_k^T y_k)^2}{s_k^T y_k} > 0$$

which concludes the proof of sufficiency.

Necessity Suppose now that $z^T H_{k+1} z > 0$ for all $z \in R^n$, $z \neq 0$. Let us choose, in particular, $z = y_k$. As H_{k+1} must satisfy the Quasi-Newton condition $H_{k+1} y_k = s_k$ we can write:

$$0 < y_k^T H_{k+1} y_k = y_k^T s_k,$$

and hence the assertion is proved. \square

If we consider the DFP algorithm

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

we can impose conditions on α_k in a way that $s_k^T y_k > 0$. In fact, this condition is equivalent to:

$$d_k^T \nabla f(x_{k+1}) > d_k^T \nabla f(x_k),$$

a condition that can be satisfied by imposing Wolfe conditions in the line search.

An important class of updating formulae that includes the DFP formula as a special case is the *Broyden class*, defined by the inverse formula:

$$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi v_k v_k^T, \quad (15.19)$$

where $\phi \geq 0$ and

$$v_k = \left(y_k^T H_k y_k \right)^{1/2} \left(\frac{s_k}{s_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k} \right). \quad (15.20)$$

From (15.19) letting $\phi = 0$ we obtain the DFP formula. For $\phi = 1$ we obtain the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) formula.

The BFGS formula can be put into the form

$$H_{k+1} = H_k + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} - \frac{s_k y_k^T H_k + H_k y_k s_k^T}{s_k^T y_k}, \quad (15.21)$$

that corresponds to the direct formula

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}. \quad (15.22)$$

We can note that the direct formula has a structure similar to that of the inverse DFP formula, which can be obtained from (15.22) by replacing B_k with H_k , and changing s_k into y_k and conversely. We note also that we can write

$$H_{k+1}^{(\text{BFGS})} = H_{k+1}^{(\text{DFP})} + v_k v_k^T, \quad (15.23)$$

where v_k is defined by (15.20), and that the matrices of the Broyden class can be defined by letting

$$H_{k+1}^{(\text{Broyden})} = (1 - \phi) H_{k+1}^{(\text{DFP})} + \phi H_{k+1}^{(\text{BFGS})}.$$

By Proposition 15.2 every formula of the Broyden class guarantees that the updated matrix remains positive definite provided that $s_k^T y_k > 0$ and $\phi \geq 0$.

Early papers on Quasi-Newton methods were mainly concerned with finite termination in the quadratic case, for functions of the form $f(x) = 1/2x^T Qx - c^T x$, with Q symmetric and positive definite.

It can be shown (see, for instance [16, p. 154]) that methods of the Broyden class with the optimal step-size, that is $\alpha_k = -\nabla f(x_K)^T d_k / d_k^T Q d_k$, converge, in n steps at most, to the minimum point of f and we have $B_n = Q$. In particular, the points generated by these methods with $H_0 = I$ coincide with the points produced by the conjugate gradient method, since the search directions constructed by the methods considered are mutually conjugate. For additional results on finite termination of rank one and rank two Quasi-Newton methods see, for instance, the book [246].

In the non quadratic case it has been shown [81] that all the methods of the Broyden class generate exactly the same points, under the assumptions that the line search along the search direction d_k is *perfect* in the sense that, for all k , we have:¹

$$\alpha_k = \min \left\{ \alpha > 0 : \nabla f(x_k + \alpha d_k)^T d_k = 0 \right\}. \quad (15.24)$$

¹ If (15.24) holds and H_k is positive definite, then $y_k^T s_k > 0$ and hence, by Proposition 15.2 the matrix H_{k+1} is positive definite.

See, for instance [231]. This result indicates that the differences between the various formulae (both in theory and in computation) are essentially due to the specific (inexact) line search adopted along the search direction or to the effects of numerical errors.

Computational experience has indicated that the BFGS formula appears to be more efficient, in comparison with the DFP formula and to the other alternatives, and hence, most of the research works, after the introduction of the BFGS method, has been devoted to the study of the theoretical properties, computational implementations and variations of this technique.

Here we report the basic scheme used in the implementations of the BFGS method, which makes use of the weak Wolfe line search and can be described by the following basic model algorithm.

BFGS Method

Data: $x_0 \in R^n$, B_0 symmetric positive definite, $\sigma \in (0, 1)$, $0 < \gamma < \min\{\sigma, 1/2\}$.

Set $k = 0$.

While $\nabla f(x_k) \neq 0$

 Set $d_k = -B_k^{-1}\nabla f(x_k)$.

 Starting from the initial stepsize $\alpha = 1$, compute α_k such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k g_k^T d_k, \quad (15.25)$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k. \quad (15.26)$$

 Set

$$x_{k+1} = x_k + \alpha_k d_k;$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k), \quad s_k = x_{k+1} - x_k.$$

 Compute

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}.$$

 Set $k = k + 1$.

End While

We note that when direct formulae are employed we must solve the system

$$B_k d = -\nabla f(x_k),$$

and this could be performed, for instance, using Cholesky factorization (possibly modified in case of ill conditioning) and updating the factorization at each step. As an alternative, we can use the inverse formula, which can also be written in the form:

$$H_{k+1} = \frac{s_k s_k^T}{s_k^T y_k} + \left(I - \frac{s_k y_k^T}{s_k^T y_k} \right) H_k \left(I - \frac{y_k s_k^T}{s_k^T y_k} \right). \quad (15.27)$$

In this case the search direction will be $d_k = -H_k \nabla f(x_k)$. Some basic properties of the BFGS method with the Wolfe line search are illustrated in the next sections.

15.4 Global Convergence of the BFGS Method

In the non quadratic case, global convergence of the BFGS algorithm with Wolfe line search has been established by Powell [215] in the convex case. In particular, we can state the following result.

Proposition 15.3 (Global Convergence in the Convex Case) *Let $f : R^n \rightarrow R$ be a convex function twice continuously differentiable on an open convex set \mathcal{D} containing the level set \mathcal{L}_0 and suppose that \mathcal{L}_0 is compact. Let $\{x_k\}$ be an infinite sequence generated by the BFGS method with Wolfe line search where $\sigma \in (0, 1)$, $0 < \gamma < \min\{\sigma, 1/2\}$. Assume that $\nabla f(x_k) \neq 0$ for all k ; then every limit point of $\{x_k\}$ is a global minimum point of f .* \square

In particular, it can be observed that the convergence result given in Proposition 15.3 depends essentially on the fact that the convexity assumption implies the validity of the following inequality for all k and for some $M > 0$

$$\frac{\|y_k\|^2}{s_k^T y_k} \leq M. \quad (15.28)$$

Thus, the assertion of Proposition 15.3 holds also for non convex functions, provided that (15.28) is satisfied. In the general, non convex case, if validity of (15.28) cannot be established, the global convergence of the (unmodified) BFGS method has not been demonstrated. We note that the model algorithm given in the preceding section is well defined and that, under the assumption that the level set \mathcal{L}_0 is compact,

the sequence $\{x_k\}$ has limit points and the sequence $\{f(x_k)\}$ converges. Moreover, recalling the properties of Wolfe line search, we know that

$$\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0.$$

Using this limit, if the eigenvalues of B_k were uniformly bounded, the global convergence of the BFGS method with Wolfe line search could be easily established, but this is apparently impossible. On the contrary, it has been shown in [58], that if the parameter γ in Wolfe line search is not too high (say $\gamma < 9 \times 10^{-3}$) and $n \geq 2$ then there exist a starting point x_0 and a C^∞ function $f : R^n \rightarrow R$ such that the sequence of gradient norms $\{\|\nabla f(x_k)\|\}$ is bounded away from zero. Then, if a convergence result for the (unmodified) BFGS method can be found, this should depend on a suitable line search algorithm.

In any case we can guarantee the global convergence along the same lines followed in the case of Newton's method, by restarting the algorithm along the negative gradient direction when required, or by modifying the BFGS formula, or else by resorting to a trust region approach.

15.5 Convergence Rate

One of the most important features of the BFGS algorithm is that, under suitable assumptions, we can establish a superlinear convergence rate and hence the method, which does not require the computation of second order derivatives, appears to be competitive in terms of convergence speed with Newton-type methods. The proof of superlinear convergence is quite long and requires establishing various preliminary results. In particular, the proof is based on the, *Dennis-Moré condition* [66], which yields necessary and sufficient conditions for the superlinear convergence of Newton-type and Quasi-Newton methods. Here we will confine ourselves to state the conclusion of this analysis.

Proposition 15.4 (Superlinear Convergence of the BFGS Method) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and let $\{x_k\}$ be the sequence produced by the BFGS method.*

Suppose that $\nabla f(x_k) \neq 0$ for all k and that $\{x_k\}$ converges to x^ , where $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite.*

Let \mathcal{C} be a convex neighborhood of x^ and assume that:*

(a) *there exist positive numbers m, M such that*

$$m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2, \quad \text{for all } x \in \mathcal{C} \text{ and } z \in R^n;$$

(continued)

Proposition 15.4 (continued)

(b) *there exists $L > 0$ such that*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|, \quad \text{for all } x, y \in \mathcal{C}.$$

Suppose further that we assume $\alpha = 1$ as initial tentative step-size in Wolfe line search. Then $\{x_k\}$ converges to x^ at a Q -superlinear convergence rate.*

□

15.6 Exercises

15.1 Define a computer code for the application of the DFP method (with exact line searches and inverse formula) to the minimization of a strictly convex quadratic function. Check that H_{n+1} is the inverse of the Hessian matrix.

15.2 Define a computer code based on the BFGS method, with (weak) Wolfe line search and perform some experiment on a set of convex and non convex test problems.

15.7 Notes and References

Quasi-Newton methods have been introduced as (variable metric methods) by Davidon in [59] (later published in [60]). Davidon's method became well known after the paper [95], which defined the method now called as DFP method. The literature on Quasi-Newton methods is very ample and all introductory books on nonlinear optimization, devote a chapter on these techniques. See, for instance, [67], [196]. We mention here some “classic” references, such as [66], [215], [38] and some works on the scaling of the terms of the Quasi-Newton formulae; see, for instance, [198], [197], [246].

Chapter 16

Methods for Nonlinear Equations



In this chapter we consider solution methods for nonlinear equations, such as Newton type methods, Quasi-Newton methods and fixed point methods, based on the use of optimization techniques. In particular, we describe globalization algorithms that take into account the structure of the problem. Non monotone globalization techniques will be defined later.

16.1 Preliminaries

Let $F : R^n \rightarrow R^n$ be a vector function with components $F_i : R^n \rightarrow R$, for $i = 1, \dots, n$, that is:

$$F(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_n(x) \end{pmatrix}.$$

We suppose that F is continuously differentiable on R^n and we denote by J the Jacobian matrix, defined by:

$$J(x) = \nabla F(x)^T = \begin{pmatrix} \nabla F_1(x)^T \\ \nabla F_2(x)^T \\ \vdots \\ \nabla F_n(x)^T \end{pmatrix}.$$

We consider the problem of solving the system

$$F(x) = 0, \tag{16.1}$$

that is the problem of finding (if exists) a point $x^* \in R^n$ such that $F(x^*) = 0$.

We are interested, in particular, in the use of optimization algorithms for solving system (16.1).

A first possibility could be that of constructing a function $f : R^n \rightarrow R$ such that $\nabla f(x) \equiv F(x)$ for all x . When this is possible and the existence of a minimizer of f can be guaranteed, the system of equations can be solved by employing algorithms for the computation of stationary points of f . An example could be the solution of a linear system, where $F(x) \equiv Qx - c = 0$, with Q symmetric and positive definite. As we already know, the problem can be solved by minimizing the quadratic function $f(x) = 1/2x^T Qx - c^T x$, whose gradient is $\nabla f(x) = Qx - c$.

In the nonlinear case, it can be shown that a continuously differentiable function F is a gradient mapping if and only if the Jacobian matrix J is symmetric. In this case a function $f : R^n \rightarrow R$ having F as gradient can be expressed in the form:

$$f(x) = \int_0^1 (x - x^0)^T F(x^0 + t(x - x^0)) dt,$$

where x^0 is an arbitrary point [200]. This *symmetry principle* restricts considerably the class of non linear problems that can be solved using the *variational approach* described above. However, as we already know, the problem of solving system (16.1) can be transformed into the problem of minimizing the function $f : R^n \rightarrow R$ defined by

$$f(x) = c \|F(x)\|^\alpha,$$

where $\|\cdot\|$ is a norm on R^n and $c, \alpha > 0$. In the sequel, unless otherwise stated, we will assume that the norm is the Euclidean norm.

We note that there exists a point $\tilde{x} \in R^n$ such that $f(\tilde{x}) = 0$ if and only if the system of equations has a solution and, in this case, \tilde{x} is obviously both a solution of the system and a global minimizer of f . If f has a global minimizer \tilde{x} such that $f(\tilde{x}) > 0$ we can exclude that the system of equations has a solution and \tilde{x} is only a point that minimizes the residual.

Thus, solving the system $F(x) = 0$ (or establishing that no solution exists) is essentially equivalent to solving a global optimization problem with an “artificial” objective function. The algorithms studied in the preceding chapters can be employed for solving system (16.1), only under suitable assumptions, which must imply that f is zero at stationary points. More specifically, assuming $f(x) = 1/2\|F(x)\|^2$, sufficient conditions that permit the use of these algorithms are the following:

- (i) there exists $x_0 \in R^n$ such that the set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact;
- (ii) the Jacobian matrix $J(x)$ is non singular at each stationary point of f in \mathcal{L}_0 :

In fact, condition (i) implies that f has a global minimizer in \mathcal{L}_0 , where ∇f is zero; condition (ii) guarantees that if $\nabla f(x) \equiv J(x)^T F(x) = 0$, then we have $F(x) = 0$.

Under these assumptions, which imply the existence of solutions to (16.1), we can adopt any globally convergent algorithm for the unconstrained minimization of continuously differentiable functions.

However, in this chapter we will restrict our attention to minimization algorithms that take explicitly into account the specific structure of the problem. In particular, we will consider the following two classes of methods.

- Methods that require the knowledge of the Jacobian matrix, that is: *Newton-type methods*, both *exact* and *inexact*.
- *Jacobian-free methods*, which do not require the computation of the Jacobian matrix, that is: *finite-difference* Newton-type methods, *Broyden's method* and *residual based* methods.

16.2 Newton-Type Methods

We already know that the (*exact*) Newton's method for the solution of the system $F(x) = 0$ is defined by the iteration

$$x_{k+1} = x_k + d_k^N, \quad (16.2)$$

where d_k^N is the solution of the system.

$$J(x_k)d + F(x_k) = 0. \quad (16.3)$$

Under the assumptions that there exists a solution x^* and that $J(x)$ is continuous and non singular in a spherical neighborhood of x^* , it has been shown that the method is locally convergent to x^* with Q -superlinear convergence rate and that the convergence rate is quadratic if the Jacobian matrix is Lipschitz-continuous.

In the *inexact Newton's* method, we have:

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

where d_k is an approximate solution of (16.3), which satisfies a condition of the form

$$\|J(x_k)d_k + F(x_k)\| \leq \eta_k \|F(x_k)\|, \quad (16.4)$$

where $\eta_k > 0$ is the *forcing term*.

Under the same assumptions stated for the exact version, it can be shown that the method converges:

- with superlinear convergence rate if η_k converges to zero;

- with quadratic convergence rate if J is Lipschitz-continuous and there exists $C > 0$ such that $\eta_k \leq C\|F(x_k)\|$.

More specifically we can state the following proposition, whose proof can be found, for instance, in [34],[152].

Proposition 16.1 (Local Convergence of Inexact Newton Method) *Let $F : R^n \rightarrow R^n$ be continuously differentiable on an open set $\mathcal{D} \subseteq R^n$. Suppose that the following conditions hold:*

- (i) *there exists a vector $x^* \in \mathcal{D}$ such that $F(x^*) = 0$;*
- (ii) *the Jacobian matrix $J(x^*)$ is non singular.*

Then, there exist an open ball $\mathcal{B}(x^; \varepsilon) \subset \mathcal{D}$, and a value $\bar{\eta}$ such that, if $x_0 \in \mathcal{B}(x^*; \varepsilon)$ and $\eta_k \in [0, \bar{\eta}]$ for all k , then the sequence $\{x_k\}$ produced by the inexact Newton method and defined by the iteration $x_{k+1} = x_k + d_k$, where d_k satisfies the condition*

$$\|J(x_k)d_k + F(x_k)\| \leq \eta_k\|F(x_k)\|, \quad (16.5)$$

converges to x^ with (at least) linear convergence rate. Moreover,*

- (a) *if $\eta_k \rightarrow 0$ then $\{x_k\}$ converges with superlinear convergence rate;*
- (b) *if the Jacobian matrix J is Lipschitz-continuous on \mathcal{D} , and there exists a constant $C > 0$ such that $\eta_k \leq C\|F(x_k)\|$ for all k , then $\{x_k\}$ converges with (at least) quadratic convergence rate.* \square

We note that in the solution of nonlinear equations we can obtain quadratic convergence, using only first order derivatives, while Newton's method for the minimization of general nonlinear functions we must make use of second order derivatives. In order to define a globalization strategy, we choose, typically, a *merit function* for deciding when the current tentative solution must be updated.

A first possibility can be that of employing the continuously differentiable function considered in the preceding section, that is, $f(x) = 1/2\|F(x)\|^2$, which is continuously differentiable. We note, in particular, that if we choose the search direction $d_k = -J(x_k)^{-1}F(x_k)$ and $F(x_k) \neq 0$, then d_k is a descent direction for f since we have.

$$\nabla f(x_k)^T d_k = -F(x_k)^T J(x_k) J(x_k)^{-1} F(x_k) = -\|F(x_k)\|^2 < 0.$$

Thus, globalization strategies already considered for Newton's method, based on line search techniques or on a trust region approach, can be easily adapted to this case, under the assumption stated in the preceding section and will not be analyzed here. In the sequel we describe an inexact Newton type method, where:

- the merit function is the (non differentiable) function $f(x) = \|F(x)\|$, and $\|\cdot\|$ is any norm;
- at each step k an Armijo-type line search is performed, along an inexact Newton's direction such that: $\|J(x_k)d_k + F(x_k)\| \leq \eta_k \|F(x_k)\|$, where $\eta_k \in (0, 1)$.

An inexact Newton scheme can be the following algorithm model.

Inexact Newton's Algorithm

Data: starting point $x^0 \in R^n$, $\gamma \in (0, 1)$, $\theta \in (0, 1)$.

For $k = 0, 1, \dots$

If $\|F(x_k)\| = 0$ terminate and exit

- (a) Choose $\eta_k \in (0, 1 - \gamma)$, compute d_k such that

$$\|J(x_k)d_k + F(x_k)\| \leq \eta_k \|F(x_k)\|.$$

- (b) Set $\alpha = 1$.

While $\|F(x_k + \alpha d_k)\| > (1 - \gamma\alpha)\|F(x_k)\|$
set $\alpha = \theta\alpha$;

End While

- (c) Set $\alpha_k = \alpha$, $x_{k+1} = x_k + \alpha_k d_k$.

End For

The assumptions required and the convergence properties of this algorithm are given in the following proposition without proof.

In Chap. 24 we will prove convergence of a non monotone globalization algorithm that includes next result as a special case.

Proposition 16.2 (Convergence of Inexact Newton's Algorithm) *Let $F : R^n \rightarrow R^n$ be continuously differentiable on R^n . Suppose that the level set $\mathcal{L}_0 = \{x \in R^n : \|F(x)\| \leq \|F(x_0)\|\}$ is compact and that there exists an open convex set Ω containing \mathcal{L}_0 , where J is non singular and Lipschitz continuous with Lipschitz constant L_J . Moreover, assume there exists $m_J > 0$ such that $\|J^{-1}(x)\| \leq m_J$ for all $x \in \Omega$. Then:*

- (a) *the inexact Newton's algorithm is well defined;*
- (b) *either the algorithm terminates for some k or the sequence $\{x_k\}$ generated by the algorithm has limit points and every limit point is a solution of the system $F(x) = 0$. \square*

We note that at Step (a) we must compute an approximate solution of the linear system

$$J(x_k)d = -F(x_k),$$

where $J(x_k)$ can be, in general, non symmetric and indefinite.

An iterative technique for solving this kind of systems is the *GMRES (Generalized Minimum RESidual)* method [232]. Under the assumption that the matrix $J(x_k)$ is non singular, the GMRES method can compute a solution of the system without requiring the explicit formation of the matrix J , using only products of $J(x)$ by a vector. This can be useful in the solution of large sparse systems and has also the advantage that the product $J(x)v$ can be approximated by finite differences.

In fact, we have

$$J(x_k)v \approx [F(x_k + \sigma v) - F(x_k)] / \sigma, \quad (16.6)$$

with $\sigma > 0$. When we use the approximation (16.6), the Inexact-Newton method is a *Jacobian-free* Newton-type method.

16.3 Broyden's Method

We already know that Quasi-Newton methods can be employed for solving systems of non linear equations $F(x) = 0$, where $F : R^n \rightarrow R^n$. In this case a Quasi-Newton algorithm can be described through the iteration

$$x_{k+1} = x_k - \alpha_k B_k^{-1} F(x_k), \quad (16.7)$$

or, equivalently: $x_{k+1} = x_k - \alpha_k H_k F(x_k)$, where B_k and H_k are approximations, respectively, of $J(x_k)$ and of $J(x_k)^{-1}$, that are updated in a way that the matrices B_{k+1} and H_{k+1} satisfy the *Quasi-Newton equations*

$$B_{k+1}(x_{k+1} - x_k) = F(x_{k+1}) - F(x_k)$$

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

One of the best known Quasi Newton methods for the solution of nonlinear equations is the Broyden's method that uses the updating formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k}, \quad (16.8)$$

where $y_k = F(x_{k+1}) - F(x_k)$ e $s_k = x_{k+1} - x_k$. The method is formally described below.

Broyden's Method

Data: starting point $x_0 \in R^n$, nonsingular matrix B_0 .

For $k = 0, 1, \dots$

- (a) Compute a direction d_k such that

$$B_k d_k = -F(x_k).$$

- (b) Set $x_{k+1} = x_k + d_k$, $y_k = F(x_{k+1}) - F(x_k)$, $s_k = x_{k+1} - x_k$ and define B_{k+1} by (16.8).

End For

Under suitable assumptions the method is *locally convergent* with superlinear convergence rate. See, for instance [152]. In particular, the following result holds.

Proposition 16.3 (Local Convergence of Broyden's Method) *Let $F : R^n \rightarrow R^n$ be continuously differentiable on an open set $\mathcal{D} \subseteq R^n$. Suppose that the following conditions hold:*

- (i) *there exists a point $x^* \in \mathcal{D}$ such that $F(x^*) = 0$;*
- (ii) *the Jacobian matrix $J(x^*)$ is nonsingular.*

There exist positive constants ϵ and δ such that, if

$$\|x_0 - x^*\| \leq \delta \quad \|B_0 - J(x^*)\| \leq \epsilon, \quad (16.9)$$

then the sequence $\{x_k\}$ generated by the Broyden's method is well-defined and converge towards x^ with superlinear convergence rate.* \square

Note that condition

$$\|B_0 - J(x^*)\| \leq \epsilon$$

is not easy to guarantee; furthermore, the performance of the method strongly depend on the choice of the initial matrix B_0 .

We also observe that the matrix B_k may be, in general, not sparse even if the Jacobian matrix is sparse, and this may be an issue in the solution of large-scale systems. This has motivated several works on the construction of sparse updates

in Quasi-Newton methods. Some references are given in the notes at the end of Chap. 18.

16.4 Residual Based Methods

When the Jacobian matrix is not available and the problem is large dimensional, in alternative to finite difference Newton-type methods and to some reduced memory version of Broyden's method, we can adopt, under suitable assumptions, a residual based method.

More specifically, we can define a Jacobian-free algorithm, where

- the merit function is defined as $f(x) = 1/2\|F(x)\|^2$;
- at each iteration k the search direction is the residual vector $F(x_k)$, possibly scaled with a scalar $1/\mu_k$, that is

$$d_k = -\frac{1}{\mu_k} F(x_k),$$

- the iteration is defined by

$$x_{k+1} = x_k + t_k \alpha_k d_k,$$

where $t_k \in \{-1, 1\}$ and α_k are determined through some line search techniques.

We note that, as the Jacobian matrix is not available, we cannot compute the gradient of f and hence we cannot establish whether d_k is a descent direction for f . Thus we need a derivative-free line search that can choose the sign t_k of the vector $F(x_k)$ and the step-size α_k .

Some specific algorithms for the choice of the parameters and the definition of the line search will be studied in Chap. 24. As shown there, convergence results can be established under the assumptions that the level set is compact and that $F(x) \neq 0$ implies that $F(x)^T J(x)^T F(x) \neq 0$. In particular, this implies that convergence (and hence also existence of solutions to $F(x) = 0$) can be guaranteed when $J(x)$ is positive or negative definite.

16.5 Notes and References

Basic references are the books [200], [67], [152]. Important works on the local properties of iterative methods are also [34], [64]. In addition to the books mentioned above, suggested references on monotone globalization techniques for Newton-type methods are [13], [35], [83]. Non monotone methods will be considered in Chaps. 24 and 25.

Chapter 17

Methods for Least Squares Problems



In this chapter we consider *least squares problems*, which constitute an important class of unconstrained optimization problems. First we recall some basic concepts and results on *linear least squares* problems and then we introduce the best known techniques for the solution of *non linear* least squares problems, such as the *Gauss-Newton* method and the *Levenberg-Marquardt* method. Finally, we add two short sections on incremental methods for linear and nonlinear least squares problems. We describe, in particular, the extension of *Kalman filter* to linear least squares problems defined in R^n .

Unless otherwise stated, we assume in this chapter that $\|\cdot\|$ is the Euclidean norm.

17.1 Least Squares Problems

Least squares problems consist in the minimization of the objective function

$$f(x) = \frac{1}{2} \sum_{i=1}^m r_i(x)^2,$$

where the terms $r_i : R^n \rightarrow R$ are called *residuals*.

Problems with this structure are often encountered in the construction of mathematical models on the basis of measurements, where each residual represents the error between the model output and the experimental data. Typical applications can be found in statistical data analysis and in machine learning. Suppose, as an example, that we possess a data set relative to some physical phenomenon of the form

$$\{(t_i, b(t_i)), i = 1, \dots, m\}$$

where $t_i \in R$ is the independent variable and $b(t_i) \in R$ is the measured value of the dependent variable. Suppose we want to construct a mathematical model of the form

$$y = g(t; x),$$

where $x \in R^n$ is a vector of parameters.

Then the residual, that is, the difference between the model output and the measured output, is given by:

$$r_i(x) = g(t_i; x) - b(t_i).$$

Suppose now we choose as model a polynomial of degree $n - 1$, that is

$$g(t; x) = x_1 + x_2 t + x_3 t^2 + \dots + x_n t^{n-1}. \quad (17.1)$$

We can attempt to approximate the physical process by choosing the parameters $x_i, i = 1, \dots, n$ that minimize a measure of the residuals, and hence, for instance, by solving the problem:

$$\min_{x \in R^n} \frac{1}{2} \sum_{i=1}^m (b(t_i) - g(t_i; x))^2. \quad (17.2)$$

Letting

$$b = \begin{pmatrix} b(t_1) \\ b(t_2) \\ \vdots \\ b(t_m) \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \quad A = \begin{pmatrix} 1 & t_1 & t_1^2 & \dots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & t_m^2 & \dots & t_m^{n-1} \end{pmatrix} = \begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{pmatrix}.$$

we can rewrite problem (17.2) in the form

$$\min_{x \in R^p} \frac{1}{2} \sum_{i=1}^m r_i^2(x) \equiv \frac{1}{2} \sum_{i=1}^m (a_i^T x - b_i)^2 \equiv \frac{1}{2} \|Ax - b\|^2. \quad (17.3)$$

Problem (17.3) is a *linear least squares* problem, where each residual r_i is an affine function $a_i^T x - b_i$ of the parameter vector x .

As an alternative to the linear model (17.1) we could consider a trigonometric function of the form

$$c_1 \sin(\omega_1 t) + c_2 \sin(\omega_2 t) + \dots + c_p \sin(\omega_p t), \quad (17.4)$$

where $c_1, \dots, c_p, \omega_1, \dots, \omega_p$ are $2p$ parameters to be determined on the basis of experimental data. Letting $n = 2p$ and

$$x_1 = c_1, x_2 = c_2, \dots, x_p = c_p, x_{p+1} = \omega_1, x_{p+2} = \omega_2, \dots, x_n = \omega_p,$$

we can define the optimization problem

$$\min_{x \in R^n} \frac{1}{2} \sum_{i=1}^m r_i^2(x), \quad (17.5)$$

where

$$r_i(x) = (b_i - x_1 \sin(x_{p+1} t_i) - x_2 \sin(x_{p+2} t_i) - \dots - x_p \sin(x_n t_i)).$$

Problem (17.5) is now a *non linear least squares problem*, since the residuals are non linear functions of the parameters.

Linear least squares problems are quadratic convex problems that always have a solution and that can be solved with both direct and iterative methods, as shortly indicated in the next section.

Non linear least squares problems are in general non convex and we can determine stationary points of the merit function by employing the algorithms studied in the preceding chapters. However, there are some specific solution techniques, described in the sequel, which take into account the particular structure of the problem.

17.2 Linear Least Squares Problems

As we have seen, linear least squares problems can be put into the form

$$\min_{x \in R^n} f(x) \equiv \frac{1}{2} \|Ax - b\|^2 \quad (17.6)$$

where A is a real $m \times n$ matrix, $x \in R^n$ and $b \in R^m$. We have

$$\nabla f(x) = A^T(Ax - b), \quad \nabla^2 f(x) = A^T A,$$

and hence the Hessian matrix is positive semidefinite, as

$$x^T A^T A x = \|Ax\|^2 \geq 0.$$

Thus, the objective function is a quadratic convex function and the optimality condition is expressed by the *normal equations*:

$$A^T A x = A^T b.$$

If A has rank n then the matrix $A^T A$ is positive definite (and hence non singular) and the least squares problem has the unique solution:

$$x^* = (A^T A)^{-1} A^T b.$$

However, it can be shown that the normal equations always admit a solution, irrespectively of the rank of A . To show this, we recall¹ that any vector $b \in R^m$ can be represented as

$$b = b_1 + b_2, \quad b_1 \in \mathcal{R}(A), \quad b_2 \in \mathcal{N}(A^T), \quad (17.7)$$

where A is any $m \times n$ real matrix, $\mathcal{R}(A)$ and $\mathcal{N}(A^T)$ are, respectively the range space of A and the null space of A^T , that is.

$$\mathcal{R}(A) = \{y \in R^m : y = Ax, \quad x \in R^n\}, \quad \mathcal{N}(A^T) = \{y \in R^m : A^T y = 0\}.$$

As $b_1 \in \mathcal{R}(A)$ we have $b_1 = A\bar{x}$ for some $\bar{x} \in R^n$ and therefore, pre-multiplying both members of the equality $b = b_1 + b_2$ by A^T and recalling that $A^T b_2 = 0$, we get

$$A^T b = A^T A \bar{x} + A^T b_2 = A^T A \bar{x},$$

which proves that \bar{x} is a solution of the normal equations. Thus a solution \bar{x} of the system $Ax = b_1$ is also a solution of the least squares problem.

Actually, the converse is also true. To show this, suppose that (17.7) holds and that \bar{x} is a solution of (17.6); then we have

$$A^T A \bar{x} = A^T b = A^T b_1.$$

It follows that \bar{x} is a solution of the least squares problem $\min \frac{1}{2} \|Ax - b_1\|^2$. On the other hand, as $b_1 \in \mathcal{R}(A)$, the minimum value of this problem should be zero and hence we must have also

$$A \bar{x} = b_1. \quad (17.8)$$

¹ A simple proof of this well known result of Linear Algebra has been given in Chap. 5, using optimality conditions for constrained problems.

From a computational point of view, for values of n up to the order of 10^3 , direct methods are typically employed, by performing suitable factorizations of A aimed at reducing, as much as possible, the effects of possible ill-conditioning of A . In particular, methods based on the *singular value decomposition* of A are often adopted. For greater values of n , iterative techniques must be used and one possibility is the adoption of the conjugate gradient method, with suitable preconditioning.

Iterative methods for least squares problems and for the solution of large linear systems are studied in [25], [152], [119].

In several applications it could be also convenient to add a regularization term of the form $\varepsilon \|x\|^2$, where ε is a small parameter, so that the problem becomes

$$\min_{x \in R^n} f(x) \equiv \frac{1}{2} \|Ax - b\|^2 + \frac{\varepsilon}{2} \|x\|^2,$$

and the Hessian matrix is now positive definite for any A .

17.3 Methods for Nonlinear Least Squares Problems

Nonlinear least squares problems are problems with objective function of the form

$$f(x) = \frac{1}{2} \sum_{i=1}^m r_i^2(x),$$

where at least one of the residuals is a non linear function of $x \in R^n$.

We suppose $m \geq n$ and that each residual is a twice continuously differentiable function on R^n .

We denote by $r : R^n \rightarrow R^m$ the vector function with components r_i , that is

$$r(x) = \begin{pmatrix} r_1(x) \\ r_2(x) \\ \vdots \\ r_m(x) \end{pmatrix}.$$

Using this notation the least squares problem can be written in the form

$$\min_{x \in R^n} f(x) \equiv \frac{1}{2} \|r(x)\|^2. \quad (17.9)$$

We denote, as usual, by J be the Jacobian matrix of r , that is the $m \times n$ matrix

$$J(x) = \nabla r(x)^T = \begin{pmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{pmatrix} = \left[\frac{\partial r_i}{\partial x_j} \right]_{i=1, \dots, m; j=1, \dots, n}$$

Then we have

$$\begin{aligned} \nabla f(x) &= \sum_{i=1}^m r_i(x) \nabla r_i(x) = J(x)^T r(x) \\ \nabla^2 f(x) &= \sum_{i=1}^m \nabla r_i(x) \nabla r_i(x)^T + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x) \\ &= J(x)^T J(x) + \sum_{i=1}^m r_i(x) \nabla^2 r_i(x) \\ &= J(x)^T J(x) + Q(x). \end{aligned} \tag{17.10}$$

We note that the Hessian matrix is expressed by the sum of the two terms $J(x)^T J(x)$ and $Q(x) \equiv \sum_{i=1}^m r_i(x) \nabla^2 r_i(x)$. Specific methods for least squares problems are based on the assumption that the term $J(x)^T J(x)$ is *dominant*, in comparison with $Q(x)$. This assumption appears to be justified in problems where the residuals are sufficiently small in a neighborhood of a solution. Some of the best known algorithms exploiting this assumption are the *Gauss-Newton method* and its modification known as *Levenberg-Marquardt method*, which will be described in the next sessions.

In the sequel, if $\{x_k\}$ is a sequence generated by an algorithm, we will use, when convenient, the simplified notation J_k , r_k , ∇f_k and $\nabla^2 f_k$, for indicating, respectively, $J(x_k)$, $r(x_k)$, $\nabla f(x_k)$ and $\nabla^2 f(x_k)$.

17.4 Gauss-Newton Method

The generic k -th iteration of Gauss-Newton method (in its pure form) is

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

where d_k^{gn} is a solution of the system

$$J_k^T J_k d = -J_k^T r_k. \quad (17.11)$$

We note that this search direction is an approximation to Newton's direction, obtained by neglecting the term $Q(x_k)$ in the expression of the Hessian matrix at x_k .

Potential advantages of this method can be the following:

- the approximation $\nabla^2 f_k \approx J_k^T J_k$ allows us to avoid the evaluation of the Hessian matrices of the residuals $\nabla^2 r_i$ for $i = 1, \dots, m$;
- in many cases, the term $J^T J$ is actually dominant, so that the efficiency of the method is comparable with that of Newton's method, even if the second order term $Q(x_k)$ is omitted;
- the vector d_k^{gn} is a solution of the linear least squares problem

$$\min_{d \in R^n} \|J_k d + r_k\|^2, \quad (17.12)$$

so that efficient specialized techniques can be used for the computation of d_k^{gn} . Moreover, it will be shown that a solution of this problem is a descent direction for f at x_k if $\nabla f(x_k) \neq 0$.

Local convergence properties and convergence rate depend on the effects of the omission of the term $Q(x)$.

In the next proposition, established in [67], we give, without proof, a local convergence result, in which we characterize, under suitable assumptions, the behaviour of Gauss-Newton method in a neighborhood of a stationary point x^* of f , that is of a point where

$$J(x^*)^T r(x^*) = 0.$$

In particular, we consider the case where the matrix $J(x^*)^T J(x^*)$ is positive definite, so that Gauss-Newton method is defined by the iteration

$$x_{k+1} = x_k - \left(J(x_k)^T J(x_k) \right)^{-1} J(x_k)^T r(x_k). \quad (17.13)$$

Proposition 17.1 *Let $f(x) = 1/2\|r(x)\|^2$ be twice continuously differentiable on an open convex set $D \subseteq R^n$. Suppose that the following conditions hold*

- (i) *there exists $x^* \in D$ such that $J(x^*)^T r(x^*) = 0$;*

(continued)

Proposition 17.1 (continued)

(ii) the matrix J is Lipschitz continuous on D , that is, there exists $L > 0$ such that

$$\|J(x) - J(y)\| \leq L\|x - y\| \quad \text{for all } x, y \in D;$$

(iii) there exists $\eta \geq 0$ such that

$$\| (J(x) - J(x^*))^T r(x^*) \| \leq \eta \|x - x^*\| \quad \text{for all } x \in D. \quad (17.14)$$

Let $\lambda \geq 0$ the smallest eigenvalue of $J(x^*)^T J(x^*)$. Then, if $\lambda > \eta$, there exists an open ball $B(x^*; \epsilon)$ such that, for every $x_0 \in B(x^*; \epsilon)$ the sequence generated by the Gauss-Newton iteration (17.13) is well defined, remains in $B(x^*; \epsilon)$, converges to x^* and we have

$$\|x_{k+1} - x^*\| \leq c_1 \|x_k - x^*\| + c_2 \|x_k - x^*\|^2 \quad c_1 \geq 0, c_2 > 0. \quad (17.15)$$

□

It can be shown that $c_1 = 0$ if $\eta = 0$ and $\lambda > 0$. Therefore, by (17.14) if $r(x^*) = 0$ we can take $\eta = 0$ and hence, if $\lambda > 0$ we have that the convergence rate is quadratic.

Now, as anticipated, we can show that a solution of (17.12) is a descent direction for f at x_k if $\nabla f(x_k) \neq 0$.

Proposition 17.2 Let $f(x) = 1/2\|r(x)\|^2$ be continuously differentiable on R^n . Let $x_k \in R^n$ be such that $\nabla f(x_k) \neq 0$. Then any solution of (17.12) is a descent direction for f at x_k .

Proof Let d_k be a solution of (17.12). We know that it can be written

$$r(x_k) = r_1(x_k) + r_2(x_k), \quad r_1(x_k) \in \mathcal{R}(J(x_k)), \quad r_2(x_k) \in \mathcal{N}(J(x_k)^T).$$

As $\nabla f_k = J_k^T (r_1^k + r_2^k) = J_k^T r_1^k \neq 0$, we have $r_1(x_k) \neq 0$. Now, recalling (17.8) of the preceding section, we can write

$$J(x_k)d_k = -r_1(x_k),$$

and hence it follows that

$$\begin{aligned}\nabla f(x_k)^T d_k &= r(x_k)^T J(x_k) d_k = -\|r_1(x_k)\|^2 - d_k^T J^T(x_k) r_2(x_k) \\ &= -\|r_1(x_k)\|^2 < 0,\end{aligned}$$

which concludes the proof. \square

A *globally convergent modification* of Gauss-Newton method is reported below..

Gauss-Newton Method

Data. $x_0 \in R^n$, $\mu_1 > 0$, $k = 0$.

While $\nabla f(x_k) \neq 0$ **do**

1. Compute a solution d_k of the problem

$$\min_{d \in R^n} \|J_k d + r_k\|^2. \quad (17.16)$$

2. Compute the step-size α_k along d_k using Armijo's method with initial tentative step-size $\Delta_k = 1$.
3. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$.

End While

A global convergence result can be easily established if we assume that the matrix $J^T J$ is uniformly positive definite on a compact level set.

Proposition 17.3 *Let $f(x) = 1/2\|r(x)\|^2$ be continuously differentiable on R^n . Suppose that the level set \mathcal{L}_0 is compact and let $\{x_k\}$ be the sequence generated by the Gauss-Newton method. Suppose that the matrix $J(x)^T J(x)$ is uniformly positive definite on \mathcal{L}_0 , that is, there exists $\eta > 0$ such that*

$$\lambda_{\min}(J(x)^T J(x)) \geq \eta > 0 \quad \text{for all } x \in \mathcal{L}_0. \quad (17.17)$$

Then, the sequence $\{x_k\}$ has limit points and every limit point of $\{x_k\}$ is a stationary point of f .

Proof We assume that $\{x_k\}$ is infinite, with $\nabla f(x_k) \neq 0$ for all k . Preliminarily we observe that by Proposition 17.2 we have $\nabla f_k^T d_k < 0$ and hence Armijo's method

guarantees that $f(x_{k+1}) < f(x_k)$ for all k and that $x_k \in \mathcal{L}_0$ for all k . As d_k satisfies (17.11) we can write

$$\nabla f_k^T d_k = (J_k^T d_k)^T d_k = -d_k^T J_k^T J_k d_k. \quad (17.18)$$

Now, recalling Propositions 9.2 and 10.3, we have that global convergence can be established, provided that the following conditions hold:

- (a) $\|d_k\| \geq c_1 \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|}$ with $c_1 > 0$;
- (b) $\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \geq c_2 \|\nabla f(x_k)\|$ with $c_2 > 0$.

We can write:

$$|\nabla f(x_k)^T d_k| / \|d_k\| = \|J(x_k) d_k\|^2 / \|d_k\| \leq \|J(x_k)\|^2 \|d_k\|. \quad (17.19)$$

As $\{x_k\} \subset \mathcal{L}_0$, by continuity of J and compactness of \mathcal{L}_0 we have, for all k , $\|J(x_k)\| \leq \beta$ for some $\beta > 0$. From (17.19) we obtain

$$\|d_k\| \geq \frac{1}{\beta^2} \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|},$$

that establishes (a). Thus the assumptions of Proposition 10.3 are satisfied and Armijo's method with unit initial step-size yields

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0. \quad (17.20)$$

Finally, in order to establish (b) we observe that, under the assumptions stated, we have, for all $x \in \mathcal{L}_0$:

$$\|J(x)z\|^2 = z^T J(x)^T J(x)z \geq \lambda_{\min}(J^T(x)J(x))\|z\|^2 \geq \eta\|z\|^2 \quad \text{for all } z \in R^n.$$

Thus we obtain

$$\begin{aligned} \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} &= \frac{|\nabla f(x_k)^T d_k|}{\|\nabla f(x_k)\| \|d_k\|} \|\nabla f(x_k)\| = \frac{\|J(x_k) d_k\|^2}{\|\nabla f(x_k)\| \|d_k\|} \|\nabla f(x_k)\| \\ &= \frac{d_k^T J(x_k)^T J(x_k) d_k}{\|J(x_k)^T J(x_k) d_k\| \|d_k\|} \|\nabla f(x_k)\| \geq \frac{\eta \|d_k\|^2}{\beta^2 \|d_k\|^2} \|\nabla f(x_k)\| \\ &= \frac{\eta}{\beta^2} \|\nabla f(x_k)\|, \end{aligned}$$

This establishes (b) and concludes the proof. \square

When the columns of J are not uniformly linearly independent, the convergence result established above may not hold. However, the Gauss-Newton algorithm is well defined, as system (17.11) always has a solution and Proposition 17.2 guarantees that d_k is a descent direction. As we know, this is not enough for proving global convergence and modifications of the search direction must be introduced. Gauss-Newton method is often implemented in the form

$$x_{k+1} = x_k - \alpha_k \left(J(x_k)^T J(x_k) + D_k \right)^{-1} J(x_k)^T r(x_k),$$

where α_k is the step-size computed through a suitable line search and D_k is a diagonal matrix such that $J(x_k)^T J(x_k) + D_k$ is uniformly positive definite.

In particular, D_k could be determined using a *modified Cholesky factorization*, similar to that described in the case of Newton's method for minimization.

An alternative approach could consist in choosing D_k as a multiple of the identity matrix, $\mu_k I$, with a suitable choice of $\mu_k \geq 0$. This technique, known as Levenberg-Marquardt method, will be considered in the next section.

17.5 Levenberg-Marquardt Method

The k -th iteration of Levenberg-Marquardt method is defined by

$$x_{k+1} = x_k - \left(J(x_k)^T J(x_k) + \mu_k I \right)^{-1} J(x_k)^T r(x_k), \quad (17.21)$$

where $\mu_k \geq 0$ is defined through suitable rules. The various implementations of the Levenberg-Marquardt method differ essentially for the criterion used for defining μ_k . Local convergence properties are similar to those established for Gauss-Newton method and are given in the next proposition, taken from [67], which we report without proof.

Proposition 17.4 Suppose that the assumptions of Proposition 17.1 are satisfied and let $\{\mu_k\}$ be a sequence of non negative scalars such that, for some $b \geq 0$ we have:

$$\mu_k \leq b < \lambda - \eta \quad \text{for all } k. \quad (17.22)$$

Then, there exists an open ball $B(x^*; \epsilon)$ such that, for every $x_0 \in B(x^*; \epsilon)$ the sequence generated by the Levenberg-Marquardt method (17.21) is well defined, remains in $B(x^*; \epsilon)$, converges to x^* and we have

$$\|x_{k+1} - x^*\| \leq c_1 \|x_k - x^*\| + c_2 \|x_k - x^*\|^2 \quad c_1 \geq 0, c_2 > 0. \quad (17.23)$$

(continued)

Proposition 17.4 (continued)

Moreover, if $r(x^*) = 0$ and $\mu_k = O(\|J(x_k)^T r(x_k)\|)$, the sequence $\{x_k\}$ converges to x^* with quadratic convergence rate. \square

A globally convergent modification of Levenberg-Marquardt method can be defined by setting:

$$x_{k+1} = x_k - \alpha_k (J_k^T J_k + \mu_k I)^{-1} J_k^T r_k, \quad (17.24)$$

where α_k is computed with an Armijo-type algorithm with unit initial step-size and μ_k is defined through the rule

$$\mu_k = \min\{\mu_1, \|J_k^T r_k\|\}, \quad (17.25)$$

where $\mu_1 > 0$. Under suitable assumptions, rule (17.25) is compatible with a superlinear convergence rate. A conceptual model of the Levenberg-Marquardt method is given below.

Levenberg-Marquardt Method

Data. $x_0 \in R^n$, $\mu_1 > 0$, $k = 0$.

While $\nabla f(x_k) \neq 0$ **do**

1. Set $\mu_k = \min\{\mu_1, \|J_k^T r_k\|\}$ and compute the solution d_k of the system

$$(J_k^T J_k + \mu_k I) d = -J_k^T r_k. \quad (17.26)$$

2. Compute the step-size α_k along d_k using Armijo's method with unit initial step-size.
3. Set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$.

End While

Global convergence of the Levenberg-Marquardt algorithm given above is stated in the next proposition [67].

Proposition 17.5 (Global Convergence) Let $f(x) = 1/2\|r(x)\|^2$ be continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence generated by the Levenberg-Marquardt method. Then, the sequence $\{x_k\}$ has limit points and every limit point of $\{x_k\}$ is a stationary point of f . \square

We note that computing d_k by solving system (17.26) is equivalent to solving the linear least squares problem

$$\min_{d \in R^n} \frac{1}{2} \left\| \begin{pmatrix} J_k \\ \mu_k^{1/2} I \end{pmatrix} d + \begin{pmatrix} r_k \\ 0 \end{pmatrix} \right\|^2.$$

It can be shown that the Levenberg-Marquardt algorithm given above has a superlinear convergence rate, under suitable assumptions. More specifically, the following result can be established for the case of zero residual [67].

Proposition 17.6 (Superlinear Convergence) Let $f(x) = 1/2\|r(x)\|^2$ be twice continuously differentiable on R^n . Let $\{x_k\}$ be the sequence generated by the Levenberg-Marquardt method. Suppose that this sequence converges to x^* , where $f(x^*) = 0$, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then $\{x_k\}$ has a superlinear convergence rate. \square

The Levenberg-Marquardt algorithm introduced before makes use of a line search globalization technique, similar to that introduced for the Gauss-Newton method. However, Levenberg-Marquardt method can also be related to a trust region approach. In fact, let us consider the problem

$$\begin{aligned} \min_{d \in R^n} & \frac{1}{2} \|J_k d + r_k\|^2 \\ & \|d\| \leq \Delta_k, \end{aligned} \tag{17.27}$$

where $\Delta_k > 0$. In problem (17.27) we refer to a quadratic model of the objective function $1/2\|r(x)\|^2$ defined by

$$m_k(d) = \frac{1}{2} \|r_k\|^2 + d^T J_k^T r_k + \frac{1}{2} d^T J_k^T J_k d.$$

In particular, it can be shown, assuming $\text{rank}(J_k) = n$ and using the theory of trust region problems, that the solution of Problem (17.27) is given by

$$d_k = - \left(J(x_k)^T J(x_k) + \mu_k I \right)^{-1} J(x_k)^T r(x_k), \quad (17.28)$$

where

$$\begin{aligned} \mu_k &= 0 \text{ if } \| (J_k^T J_k)^{-1} J_k^T r_k \| \leq \Delta_k; \\ \mu_k &> 0 \text{ otherwise.} \end{aligned}$$

This implies that the Levenberg-Marquardt method can be analyzed as a trust region method and hence that globally convergent techniques can be constructed by updating the parameters Δ_k and μ_k with techniques used for globalizing trust region methods.

17.6 Recursive Linear Least Squares Algorithm

In many real problems, for instance those concerning the supervised training of machine learning models, the objective function of a least squares problem is defined by the sum of a huge number of terms (corresponding to the training data). In these cases, it may be convenient to adopt an *incremental* algorithm which uses some terms (for instance, one term) of the objective function at a time rather than using the whole objective function.

We present here a version of the *Kalman filter*, originally proposed for estimating the state of dynamical systems [147], as an incremental algorithm for solving a linear least squares problem. As we will see, it is based on recursive formulae.

Consider the linear least squares problem corresponding to the following objective function:

$$f^{(k)}(x) = \frac{1}{2} \sum_{i=1}^k (a_i^T x - b_i)^2,$$

and let $x(k)$ be the optimal solution, i.e.,

$$x(k) = \underset{x}{\text{Arg}} \min \frac{1}{2} \sum_{i=1}^k (a_i^T x - b_i)^2.$$

We want to define the optimal solution of the new problem corresponding to the objective function

$$f^{(k+1)}(x) = \frac{1}{2} \sum_{i=1}^k (a_i^T x - b_i)^2 + \frac{1}{2} (a_{k+1}^T x - b_{k+1})^2,$$

obtained by adding the term $\frac{1}{2}(a_{k+1}^T x - b_{k+1})^2$ to the previous function $f^{(k)}$. Let $x(k+1)$ be the optimal solution of the new problem. Setting

$$\tilde{A} = \begin{pmatrix} a_1^T \\ \vdots \\ a_k^T \\ a_{k+1}^T \end{pmatrix} \quad \tilde{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \\ b_{k+1} \end{pmatrix}, \quad A = \begin{pmatrix} \tilde{A} \\ a_{k+1}^T \end{pmatrix} \quad b = \begin{pmatrix} \tilde{b} \\ b_{k+1} \end{pmatrix},$$

we can write

$$f^{(k)}(x) = \frac{1}{2} \|\tilde{A}x - \tilde{b}\|^2,$$

$$f^{(k+1)}(x) = \frac{1}{2} \|Ax - b\|^2.$$

Since $x(k)$ is the vector minimizing $f^{(k)}$ we have that it satisfies the normal equations

$$\tilde{A}^T \tilde{A}x(k) = \tilde{A}^T \tilde{b}, \quad (17.29)$$

while the solution $x(k+1)$ of the new problem must satisfy the following normal equations

$$A^T Ax(k+1) = A^T b.$$

Last equation can be rewritten as follows

$$\left[\begin{pmatrix} \tilde{A} \\ a_{k+1}^T \end{pmatrix}^T \begin{pmatrix} \tilde{A} \\ a_{k+1}^T \end{pmatrix} \right] x(k+1) = \begin{pmatrix} \tilde{A} \\ a_{k+1}^T \end{pmatrix}^T \begin{pmatrix} \tilde{b} \\ b_{k+1} \end{pmatrix},$$

from which, setting

$$H(k+1) = A^T A = \tilde{A}^T \tilde{A} + a_{k+1} a_{k+1}^T,$$

we obtain

$$H(k+1)x(k+1) = \tilde{A}^T \tilde{b} + a_{k+1} b_{k+1}.$$

Recalling (17.29) we have

$$\begin{aligned} H(k+1)x(k+1) &= \tilde{A}^T \tilde{A}x(k) + a_{k+1} b_{k+1} \\ &= (\tilde{A}^T \tilde{A} + a_{k+1} a_{k+1}^T)x(k) - a_{k+1} a_{k+1}^T x(k) + a_{k+1} b_{k+1} \\ &= H(k+1)x(k) + a_{k+1}(b_{k+1} - a_{k+1}^T x(k)). \end{aligned}$$

Now assume that $H(k) = \tilde{A}^T \tilde{A}$ is nonsingular. Then also $H(k+1)$ is nonsingular and we can write the following recursive updating formulae

$$x(k+1) = x(k) + H(k+1)^{-1} a_{k+1} (b_{k+1} - a_{k+1}^T x(k)),$$

where

$$H(k+1) = H(k) + a_{k+1} a_{k+1}^T, \quad k = k_0, k_0 + 1, \dots,$$

and, applying the *Sherman-Morrison-Woodbury* formula (see Proposition 15.1), we have

$$H(k+1)^{-1} = H(k)^{-1} - \frac{H(k)^{-1} a_{k+1} a_{k+1}^T H(k)^{-1}}{1 + a_{k+1}^T H(k)^{-1} a_{k+1}}.$$

Note that the above formulae do not require to form the matrix A , $(k+1) \times n$, and this is advantageous whenever the number $k+1$ of data is “bigger” than the number n of variables.

We observe that we must assume that the initial matrix $H(k_0)$ is non singular (and hence that k_0 is sufficiently large).

17.7 Some Notes on Incremental Methods for Nonlinear Problems

Consider the problem

$$\min_{x \in R^n} f(x) = \sum_{i=1}^m f_i(x), \quad (17.30)$$

where $f_i : R^n \rightarrow R$ are continuously differentiable functions over R^n . A particular case of (17.30) is a least squares problem.

Methods for solving (17.30) can be divided into two classes:

- *batch* methods, using information on the whole $f(x)$ and on its derivatives;
- *incremental* (or *online*) methods, using information of a subset (a *mini batch*) of terms $f_i(x)$ and their derivatives.

The classical methods, like gradient, conjugate gradient, Newton, Quasi-Newton methods are batch methods.

We limit ourselves to describe a simple incremental gradient method using a single term f_i at a time. Given the current point x_k , the updated point x_{k+1} is obtained at the end of the following cycle:

$$\begin{aligned} z_0 &= x_k \\ z_i &= z_{i-1} - \alpha_k \nabla f_i(z_{i-1}) \quad i = 1, \dots, m \\ x_{k+1} &= z_m, \end{aligned}$$

where $\alpha_k > 0$ is the stepsize. We omitted the dependence of z_i on k in order to simplify the notation. Two possible advantages of incremental methods can be the following:

- they can provide “good” solutions in problems where the single terms $f_i(x)$ of the objective function are not *offline* available but are generated in *real time*;
- if the number of data is huge, i.e., m is huge, it may happen that the data show *statistical homogeneity*, and hence a suitable vector could be an approximate minimum point of several term f_i , so that, a “sufficiently good” solution of the whole problem could be achieved by a single iteration of the cycle.

Note that we can write

$$x_{k+1} = x_k - \alpha_k \sum_{i=1}^m \nabla f_i(z_{i-1}). \quad (17.31)$$

Therefore, with respect to the gradient method, an incremental method replaces the steepest direction

$$-\sum_{i=1}^m \nabla f_i(x_k).$$

by the following direction

$$-\sum_{i=1}^m \nabla f_i(z_{i-1}).$$

As a consequence, an incremental method can be viewed as a *gradient method with error*. In particular, we have

$$x_{k+1} = x_k - \alpha_k (\nabla f(x_k) + e_k), \quad (17.32)$$

where

$$e_k = \sum_{i=1}^m (\nabla f_i(z_{i-1}) - \nabla f_i(x_k)). \quad (17.33)$$

The choice of the stepsize α_k is crucial for ensuring the global convergence of an incremental method. From (17.32) we get that the direction of an incremental method differs from the steepest direction by a quantity proportional to α_k . This suggests that the stepsize must be iteratively reduced to guarantee the global convergence. Furthermore, if the sequence $\{x_k\}$ converges then the vectors

$$\alpha_k \nabla f_i(z_{i-1}), \quad \text{for } i = 1, \dots, m$$

must tend to zero, and this implies that necessarily $\alpha_k \rightarrow 0$ (otherwise we would have that all the single gradients must tend to zero).

A convergence result of an incremental method viewed as *gradient method with error* is reported in the following proposition whose proof can be found in [16].

Proposition 17.7 (Convergence of an Incremental Method) *Let $f(x) = \sum_{i=1}^m f_i(x)$, where $f_i : R^n \rightarrow R$ are continuously differentiable function over R^n . Let $\{x_k\}$ be the sequence (17.31) generated by an incremental gradient method. Assume that there exist three positive constant $L, C, e D$ such that, for $i = 1, \dots, m$, we have*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L \|x - y\| \quad \text{for all } x, y \in R^n \quad (17.34)$$

$$\|\nabla f_i(x)\| \leq C + D \|\nabla f(x)\| \quad \text{for all } x \in R^n. \quad (17.35)$$

Assume that

$$\sum_{k=0}^{\infty} \alpha_k = \infty \quad \sum_{k=0}^{\infty} (\alpha_k)^2 < \infty. \quad (17.36)$$

Then, either $f(y_k) \rightarrow -\infty$, or the sequence $\{f(x_k)\}$ converges to a finite value and we have

- (i) $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$;
- (ii) every limit point of $\{x_k\}$ is a stationary point of f .

□

17.8 Exercises

17.1 Let t_1, t_2, \dots, t_{12} be a set of time instants and let $b(t_i), i = 1, \dots, 12$ be the temperature values measured at $t_i, i = 1, \dots, 12$. Suppose we would represent the time behaviour of the temperature with a model of the form

$$m(t) = c_1 e^t + c_2 e^{t^2} + c_3 e^{t-6}.$$

Formulate a least squares problem for computing the values of the parameters c_1, c_2, c_3 and discuss the problem obtained. State whether the problem is a linear or a nonlinear least squares problem.

17.2 Let J be a $m \times n$ real matrix with $m \geq n$. Show that the matrix $J^T J$ is positive definite if and only if $\text{rank}(J) = n$.

17.3 Show that computing d_k by solving system

$$(J_k^T J_k + \mu_k I) d = -J_k^T r_k$$

is equivalent to solving the linear least squares problem

$$\min_{d \in R^n} \frac{1}{2} \left\| \begin{pmatrix} J_k \\ \mu_k^{1/2} I \end{pmatrix} d + \begin{pmatrix} r_k \\ 0 \end{pmatrix} \right\|^2.$$

17.4 Define a globally convergent version of the Gauss-Newton method (towards stationary points of the error function) using a hybrid method, based on the occasional use of the gradient direction on the basis of suitable tests. Construct a computer code, where the Gauss-Newton direction is computed through the conjugate gradient method for linear least squares problems. Then perform some numerical experiments.

17.5 Define a computer code for the Levenberg-Marquardt method, where the search direction is obtained through the conjugate gradient method for solving the system

$$(J_k^T J_k + \mu_k I) d = -J_k^T r_k$$

and perform some numerical experiments.

17.9 Notes and References

Methods for linear-least squares problems are deeply analyzed in [25]. The proofs of local convergence results of the Gauss-Newton method and of the Levenberg-Marquardt method can be found in [67]. The chapter also contains some notes on incremental methods that, in recent times, received more and more attention especially in the context of machine learning from real-time data. The literature on incremental methods is very wide and would deserve a suitable space to be deeply analyzed and discussed in connection with complexity analysis issues. We limit ourselves to cite a few of relevant papers like [9], [63], [121], [156].

Chapter 18

Methods for Large-Scale Optimization



In this chapter we present methods for solving *large scale* nonlinear equations and nonlinear unconstrained optimization problems. In particular, we describe *inexact* and *truncated* Newton-type methods and we state local and global convergence results for these techniques. Finally we consider *limited memory Quasi-Newton* methods that do not require storage of large matrices and matrix operations.

Methods based on decomposition techniques will be considered in Chap. 26.

18.1 Solution Strategies for Large Scale Problems

Many applications may require to solve on R^n *large-scale* systems of nonlinear equations or nonlinear unconstrained optimization problems, i.e., nonlinear problems with a very large number of variables. Obviously, the concept of “large-scale” is related to the available computational power and to the problem structure, but actually we can suppose that a large-scale nonlinear unconstrained problem is, in general, a problem with some thousands of variables. In a large-scale problem, without sparsity assumptions, it is not realistic to store and, possibly, factorize $n \times n$ matrices (or their approximations) explicitly. Gradient methods and conjugate gradient methods can be applied, in principle, without the need of introducing modifications for taking into account the huge number of variables. Indeed, these methods are based on information on the objective function and on the gradient, and neither require matrix storage nor solution of large linear systems at each step.

Newton and Quasi-Newton methods, which would guarantee a much faster convergence speed, can not be directly applied since, as already seen, they require to store Jacobian or Hessian matrices or their approximations. Then, to overcome these limits in the solution of large-scale problems, the following classes of Newton and Quasi-Newton methods have been introduced

- *inexact and truncated Newton methods*, where the linear system underlying the computation of the Newton direction is *approximately* solved by iterative procedures that do not require the storage of matrices;
- *limited-memory Quasi-Newton methods*, which make simple approximations of the Jacobian or Hessian matrix and require minimal storage.

We remark that usually the terminology *inexact Newton* method refers to a method for solving systems of nonlinear equations (see Chap. 16), while *truncated Newton* method refers to a method for the solution of an optimization problem.

18.2 Truncated Newton Method

In order to apply the Newton method for minimizing a function $f : R^n \rightarrow R$, it is necessary to solve (whenever possible) the linear system

$$\nabla^2 f(x_k)d + \nabla f(x_k) = 0, \quad (18.1)$$

whose solution provides the search direction.

When the dimension of the problem is huge, the computation of the solution of the linear system may become too expensive. Therefore, it could be reasonable to solve approximately the linear system by an iterative procedure. An important question concerns the level of accuracy required to ensure the same fast convergence speed of the Newton method.

The iteration of an *inexact Newton* method takes the form

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

where d_k is not an exact solution of (18.1), but it is an *approximated solution*, i.e., it is such that the following condition holds

$$\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|, \quad (18.2)$$

where $\eta_k > 0$ is the so-called *forcing term*.

We will analyze the conditions on η_k which ensure that an inexact Newton method locally converges to a stationary point of f with a given convergence rate. We will see that the method converges

- with linear convergence rate provided that η_k is “sufficiently small”;
- with superlinear convergence rate provided that η_k tends to zero;
- with quadratic convergence rate provided that η_k is an $O(\|\nabla f(x_k)\|)$.

More specifically, as already seen in the analysis of convergence of the Newton method, the result stated by Proposition 16.1 in Chap. 16 can be directly extended to the case of an inexact Newton method for the computation of stationary points

of a function $f : R^n \rightarrow R$. In particular, by Proposition 16.1, we can state the following result.

Proposition 18.1 (Local Convergence of Truncated Newton) *Let $f : R^n \rightarrow R$ be a twice continuously differentiable function on an open set $\mathcal{D} \subseteq R^n$. Suppose that the following conditions hold:*

- (i) *there exists a point $x^* \in \mathcal{D}$ such that $\nabla f(x^*) = 0$;*
- (ii) *the Hessian matrix $\nabla^2 f(x^*)$ is non singular.*

Then, there exist an open ball $\mathcal{B}(x^; \varepsilon) \subset \mathcal{D}$, and a value $\bar{\eta}$ such that, if $x_0 \in \mathcal{B}(x^*; \varepsilon)$ and $\eta_k \in [0, \bar{\eta}]$ for all k , then the sequence $\{x_k\}$ generated by the inexact Newton method and defined by the iteration $x_{k+1} = x_k + d_k$, where d_k satisfies the condition*

$$\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|,$$

converges to x^ with (at least) linear convergence rate. Moreover,*

- (a) *if $\eta_k \rightarrow 0$ then $\{x_k\}$ converges with superlinear convergence rate;*
- (b) *if the Hessian matrix $\nabla^2 f$ is Lipschitz-continuous on \mathcal{D} , and there exists a constant $C > 0$ such that $\eta_k \leq C \|\nabla f(x_k)\|$ for all k , then $\{x_k\}$ converges with (at least) quadratic convergence rate.* \square

18.3 Globally Convergent Truncated Newton Methods

18.3.1 Preliminaries

In the preceding paragraph we have presented convergence results based on the computation of inexact solutions of the linear system underlying the Newton method, that is, the system

$$\nabla^2 f(x_k)d + \nabla f(x_k) = 0. \quad (18.3)$$

The conjugate gradient method can be applied, within a Newton method, for computing the search direction, that is, for (either exactly or inexactly) solving system (18.3).

Setting

$$q_k(d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d,$$

we can use the conjugate gradient method for computing d_k minimizing with respect to d the function $q_k(d)$. The conjugate gradient method can be stopped whenever the residual of the Newton equation satisfies a suitable convergence condition.

The residual of the Newton equation is given by the gradient of $q_k(d)$ with respect to d , i.e., by the vector

$$\nabla q_k(d) = \nabla f(x_k) + \nabla^2 f(x_k)d.$$

From Proposition 18.1 we get that the following condition is sufficient for ensuring a superlinear convergence rate

$$\lim_{k \rightarrow \infty} \frac{\|\nabla q_k(d_k)\|}{\|\nabla f(x_k)\|} = 0.$$

We indicate by $d^{(i)}$ the approximations of d_k generated by the conjugate gradient method. The inner iteration is typically stopped or “truncated” using the following criterion

$$\|\nabla q_k(d^{(i)})\| \leq \eta \|\nabla f(x_k)\| \min \left\{ \frac{1}{k+1}, \|\nabla f(x_k)\| \right\},$$

where $\eta > 0$ is a given constant. In this way, a relatively low level of precision is required at the early iterations, while the required level increases when the current point is close to the solution. This strategy permits to considerably reduce the computational cost for approximating the Newton direction.

However, we remark that in the general case the Hessian matrix $\nabla^2 f(x_k)$ may be not positive definite, so that, the conjugate gradient method must be suitably adapted in order to determine a descent direction.

We will show a truncated Newton method based on a line search. Truncated Newton methods based on the trust region strategy can be defined in a very similar way.

18.3.2 A Truncated Newton Method Based on a Line Search

A scheme of a *Truncated Newton* (TN) method is reported below, where, in order to simplify notation, we set

$$\nabla q^{(i)} = \nabla q_k(d^{(i)}) = \nabla f(x_k) + \nabla^2 f(x_k)d^{(i)}.$$

Notice that the vectors $d^{(i)}$ are the approximations of the Newton direction generated by the conjugate gradient method, the vectors $s^{(i)}$ are the conjugate directions (with respect to $\nabla^2 f(x_k)$) used in the inner cycle to compute $d^{(i)}$.

TN Algorithm

Data: $\eta > 0, \varepsilon > 0$

Step 1. Given $x_0 \in R^n$, set $k = 0$.

Step 2. Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ then stop; otherwise:

Step 2.1. Set $i = 0, d^{(0)} = 0, s^{(0)} = -\nabla q^{(0)} = -\nabla f(x_k)$.

Step 2.2. If $s^{(i)T} \nabla^2 f(x_k) s^{(i)} \leq \varepsilon \|s^{(i)}\|^2$ set

$$d_k = \begin{cases} -\nabla f(x_k), & \text{if } i = 0, \\ d^{(i)}, & \text{if } i > 0 \end{cases}$$

and go to Step 3.

Step 2.3. Compute:

$$\alpha^{(i)} = -\frac{\nabla q^{(i)T} s^{(i)}}{s^{(i)T} \nabla^2 f(x_k) s^{(i)}}$$

$$d^{(i+1)} = d^{(i)} + \alpha^{(i)} s^{(i)}$$

$$\nabla q^{(i+1)} = \nabla q^{(i)} + \alpha^{(i)} \nabla^2 f(x_k) s^{(i)}.$$

If $\|\nabla q^{(i)}\| \leq \eta \|\nabla f(x_k)\| \min \left\{ \frac{1}{k+1}, \|\nabla f(x_k)\| \right\}$,

then set $d_k = d^{(i)}$ and go to Step 3.

Step 2.4. Compute:

$$\beta^{(i+1)} = \frac{\nabla q^{(i+1)T} \nabla^2 f(x_k) s^{(i)}}{s^{(i)T} \nabla^2 f(x_k) s^{(i)}}$$

$$s^{(i+1)} = -\nabla q^{(i+1)} + \beta^{(i+1)} s^{(i)},$$

set $i = i + 1$ and go to Step 2.2.

Step 3. Compute the step-size α_k along d_k by the Armijo method (with unitary initial step-size), set $x_{k+1} = x_k + \alpha_k d_k$, $k = k + 1$ and go to Step 2.

We observe that the instructions of Algorithm TN imply that the Hessian $\nabla^2 f$ is always multiplied by the vector $s^{(i)}$. Hence, it is sufficient to provide a procedure that computes this matrix-vector product, without the need of explicitly storing the

matrix $\nabla^2 f$ in the working memory, and this can be convenient in the solution of large-scale problems.

If the Hessian matrix is not available, it is possible to approximate the products $\nabla^2 f(x_k)s^{(i)}$ with finite-difference formulas, by setting

$$\nabla^2 f(x_k)s^{(i)} \approx \frac{\nabla f(x_k + ts^{(i)}) - \nabla f(x_k)}{t},$$

where t is suitably chosen.

Remark 18.1 The algorithm is well-defined, that is, the cycle of Step 2 terminates in a finite number of iterations. Indeed, if for $i = 0, \dots, n-1$ the tests at Step 2.2 and Step 2.3 are never satisfied, from the properties of the conjugate gradient method it follows $\|\nabla q^{(n)}\| = 0$, and hence the test at Step 2.3 is satisfied. \square

In order to prove the convergence of Algorithm TN it is sufficient to show that the search direction is a descent direction that satisfies suitable conditions. Indeed, from known results, these conditions on the search direction and the employment of the Armijo line search for the computation of the step-size allow us to prove the global convergence of Algorithm TN.

Preliminarily we state the following result.

Proposition 18.2 Let $f : R^n \rightarrow R$ be twice continuously differentiable and assume that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by Algorithm TN. There exist two numbers $c_1 > 0$ and $c_2 > 0$ such that for all k we have

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^2 \quad (18.4)$$

$$\|d_k\| \leq c_2 \|\nabla f(x_k)\|. \quad (18.5)$$

Proof We observe that the formulas of the conjugate gradient method for the quadratic case are still valid until the iterations of the inner cycle are stopped. Note that, for the quadratic case, in the conjugate gradient method we have

$$\alpha^{(i)} = -\frac{\nabla q^{(i)T} s^{(i)}}{s^{(i)T} \nabla^2 f(x_k) s^{(i)}} = -\frac{\nabla q^{(0)T} s^{(i)}}{s^{(i)T} \nabla^2 f(x_k) s^{(i)}}.$$

This follows from the fact that we can write

$$\nabla q^{(i)} = \nabla q^{(0)} + \sum_{j=0}^{i-1} \alpha^{(j)} \nabla^2 f(x_k) s^{(j)},$$

for which, multiplying for $s^{(i)}$ and taking into account that the vectors $s^{(j)}$ are mutually conjugate, we obtain:

$$s^{(i)T} \nabla q^{(i)} = s^{(i)T} \nabla q^{(0)}.$$

Now let d_k be the direction computed by Algorithm TN. By construction, either $d_k = -\nabla f(x_k)$ or $d_k = d^{(i)}$ (for a suitable value of the index i). In the second case we can write

$$d_k = d^{(i)} = \sum_{j=0}^{i-1} \alpha^{(j)} s^{(j)} = - \sum_{j=0}^{i-1} \frac{\nabla q^{(0)T} s^{(j)}}{s^{(j)T} \nabla^2 f(x_k) s^{(j)}} s^{(j)},$$

from which, recalling that in the algorithm we have

$$\nabla q^{(0)} = \nabla f(x_k), \quad s^{(0)} = -\nabla f(x_k),$$

we obtain

$$\begin{aligned} \nabla f(x_k)^T d_k &= - \sum_{j=0}^{i-1} \frac{(\nabla f(x_k)^T s^{(j)})^2}{s^{(j)T} \nabla^2 f(x_k) s^{(j)}} \\ &\leq - \frac{(\nabla f(x_k)^T \nabla f(x_k))^2}{\nabla f(x_k)^T \nabla^2 f(x_k) \nabla f(x_k)}. \end{aligned}$$

It follows that $\nabla f(x_k)^T d_k < 0$, furthermore we can write

$$|\nabla f(x_k)^T d_k| \geq \frac{\|\nabla f(x_k)\|^4}{\|\nabla f(x_k)\|^2 \|\nabla^2 f(x_k)\|} \geq \frac{1}{M} \|\nabla f(x_k)\|^2,$$

where $M > 0$ is an upper bound of $\|\nabla^2 f(x_k)\|$ on the level set. Taking into account that we can have $d_k = -\nabla f(x_k)$, we can conclude that (18.4) is satisfied with $c_1 \leq \min\{1, 1/M\}$. As regards (18.5), if $d_k = d^{(i)}$ then we have

$$\|d_k\| = \|d^{(i)}\| \leq \sum_{j=0}^{i-1} \left| \frac{s^{(j)T} s^{(j)}}{s^{(j)T} \nabla^2 f(x_k) s^{(j)}} \right| \|\nabla f(x_k)\|,$$

and hence, since we must have

$$s^{(j)T} \nabla^2 f(x_k) s^{(j)} > \varepsilon \|s^{(j)}\|^2$$

(see the instructions of Step 2.2), we obtain

$$\|d^{(i)}\| \leq \frac{i}{\varepsilon} \|\nabla f(x_k)\| \leq \frac{n}{\varepsilon} \|\nabla f(x_k)\|.$$

Then, condition (18.5) holds in any case setting:

$$c_2 = \max\{1, \frac{n}{\varepsilon}\}.$$

□

Now we can state the following global convergence result.

Proposition 18.3 Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and suppose that the level set \mathcal{L}_0 is compact. The sequence generated by Algorithm TN admits limit points and every limit point is a stationary point of f .

Proof Proposition 18.2 ensures that d_k is a descent direction. The employment of the Armijo method at Step 3 implies $f(x_{k+1}) < f(x_k)$. Then, (18.4), (18.5), the properties of the Armijo method (see Propositions 10.3 and 9.2) that the sequence $\{x_k\}$ admits limit points and every limit point is a stationary point of f . □

Under suitable assumptions on the Hessian it is possible to prove superlinear and quadratic convergence rate of Algorithm TN. Preliminarily we recall a known result on the acceptance of the unitary step-size by the Armijo rule.

Proposition 18.4 (Acceptance of the Unit Step-Size) Let $f : R^n \rightarrow R$ twice continuously differentiable on R^n and let $\{x_k\}$ be a sequence generated by a method defined by an iteration of the form

$$x_{k+1} = x_k + \alpha_k d_k.$$

Suppose that $\{x_k\}$ converges to x^* , where $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Assume that $\nabla f(x_k) \neq 0$ for all k and

$$\lim_{k \rightarrow \infty} \frac{\|d_k + \nabla^2 f(x^*)^{-1} \nabla f(x_k)\|}{\|\nabla f(x_k)\|} = 0. \quad (18.6)$$

(continued)

Proposition 18.4 (continued)

Then, if $\gamma \in (0, 1/2)$, there exists an index k^* , such that, for all $k \geq k^*$ we have

$$f(x_k + d_k) \leq f(x_k) + \gamma \nabla f(x_k)^T d_k,$$

that is, the unit step-size is accepted by the Armijo rule. Moreover we have

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \quad (18.7)$$

Remark 18.2 We observe that condition (18.6) is satisfied by a direction d_k such that

$$\|\nabla^2 f(x_k) d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|,$$

where $\eta_k \rightarrow 0$. Indeed, for k sufficiently large we can write

$$\begin{aligned} \eta_k &\geq \frac{\|\nabla^2 f(x_k) d_k + \nabla f(x_k)\|}{\|\nabla f(x_k)\|} = \frac{\|\nabla^2 f(x_k)^{-1}\| \|\nabla^2 f(x_k) d_k + \nabla f(x_k)\|}{\|\nabla^2 f(x_k)^{-1}\| \|\nabla f(x_k)\|} \\ &\geq \frac{\|d_k + \nabla^2 f(x_k)^{-1} \nabla f(x_k)\|}{\|\nabla^2 f(x_k)^{-1}\| \|\nabla f(x_k)\|}, \end{aligned}$$

from which it follows

$$\frac{\|d_k + \nabla^2 f(x_k)^{-1} \nabla f(x_k)\|}{\|\nabla f(x_k)\|} \leq \eta_k \|\nabla^2 f(x_k)^{-1}\|.$$

Taking into account that

$$\begin{aligned} \frac{\|d_k + \nabla^2 f(x^*)^{-1} \nabla f(x_k)\|}{\|\nabla f(x_k)\|} &\leq \frac{\|d_k + \nabla^2 f(x_k)^{-1} \nabla f(x_k)\|}{\|\nabla f(x_k)\|} \\ &+ \frac{\|\nabla^2 f(x^*)^{-1} - \nabla^2 f(x_k)^{-1}\| \|\nabla f(x_k)\|}{\|\nabla f(x_k)\|}, \end{aligned}$$

we can conclude that (18.6) holds. \square

We can prove the superlinear and quadratic convergence rate of the truncated Newton method.

Proposition 18.5 Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n . Suppose that the sequence $\{x_k\}$ generated by Algorithm TN converges to x^* , where $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite. Then:

- (i) there exists a value $\bar{\epsilon} > 0$ such that, for k sufficiently large and for any $\epsilon \in (0, \bar{\epsilon}]$, the test at Step 2.2 is never satisfied;
- (ii) the sequence $\{x_k\}$ converges to x^* with superlinear convergence rate;
- (iii) if the Hessian matrix $\nabla^2 f$ is Lipschitz-continuous in a neighborhood of x^* , the sequence $\{x_k\}$ converges to x^* with quadratic convergence rate.

Proof

- (i) The convergence of $\{x_k\}$, the continuity of the Hessian and the assumption that $\nabla^2 f(x^*)$ is positive definite imply that for k sufficiently large

$$\lambda_m[\nabla^2 f(x_k)] \geq \bar{\lambda} > 0,$$

where $\lambda_m[\nabla^2 f(x_k)]$ is the minimum eigenvalue of $\nabla^2 f(x_k)$. Set $\bar{\epsilon} = \frac{\bar{\lambda}}{4}$, and let $\epsilon \in (0, \bar{\epsilon}]$. For k sufficiently large we can write

$$s^{(i)T} \nabla^2 f(x_k) s^{(i)} \geq \frac{\bar{\lambda}}{2} \|s^{(i)}\|^2 > \epsilon \|s^{(i)}\|^2,$$

and hence assertion (i) is proved.

- (ii) For k sufficiently large the matrix $\nabla^2 f(x_k)$ is positive definite, so that, from the properties of the conjugate gradient method it follows that the test at Step 2.3 is satisfied. Hence we can write

$$\|\nabla^2 f(x_k) d_k + \nabla f(x_k)\| \leq \eta \|\nabla f(x_k)\| \min\{1/(k+1), \|\nabla f(x_k)\|\}.$$

Therefore, we have

$$\|\nabla^2 f(x_k) d_k + \nabla f(x_k)\| \leq \eta_k \|\nabla f(x_k)\|, \quad (18.8)$$

with $\eta_k \rightarrow 0$, being $\eta_k = \min\{1/(k+1), \|\nabla f(x_k)\|\}$. Condition (18.6) of Proposition 18.4 is satisfied (see the remark after the proposition), and hence we have

$$x_{k+1} = x_k + d_k, \quad (18.9)$$

where d_k satisfies (18.8). Then, Proposition 18.1 implies that $\{x_k\}$ converges to x^* with superlinear convergence rate.

- (iii) The assertion follows from (18.8), (18.9) and Proposition 18.1, recalling that $\eta_k \leq \eta \|\nabla f(x_k)\|$.

□

18.4 Quasi-Newton Methods for Large-Scale Optimization

18.4.1 Preliminaries

We have already seen that the BFGS method is defined by an iteration of the form

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

where α_k is the step-size computed by a line search based on the Wolfe conditions and the matrix H_k satisfies the equation

$$H_k y_k = s_k,$$

with

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

The matrix H_k is an approximation of the inverse of the Hessian $\nabla^2 f(x_k)$ and is updated using the pair $\{s_k, y_k\}$ according the formula

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \quad (18.10)$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \quad V_k = I - \rho_k y_k s_k^T. \quad (18.11)$$

The matrix H_k is usually dense, so that, its storage as well as the computational cost of the matrix operations may be prohibitive when the number of variables is huge. In order to overcome these limits, suitable modifications of Quasi-Newton methods have been proposed:

- *memoryless Quasi-Newton* methods;
- *limited-memory Quasi-Newton* methods (*limited-memory BFGS, L-BFGS*).

18.4.2 Memoryless Quasi-Newton Methods

The idea underlying these methods is to perform at each iteration k a “reset” of the approximation H_k of the inverse of the Hessian setting $H_k = I$ in the updating formula BFGS given by (18.10), that is, defining the formula

$$H_{k+1} = V_k^T V_k + \rho_k s_k s_k^T. \quad (18.12)$$

The methods using as search direction the vector

$$d_{k+1} = -H_{k+1} \nabla f(x_{k+1}),$$

where H_{k+1} is defined by (18.12), are called *memoryless Quasi-Newton* methods, since they require at each iteration k only to store the pair of vectors (s_k, y_k) . There are in the literature several formulas (18.12) that will be not analyzed in this book.

It is interesting to analyze the relationship between memoryless Quasi-Newton methods and conjugate gradient methods. We can show that a Quasi-Newton method based on (18.12) and using an exact line search corresponds to the Hestenes-Stiefel conjugate gradient method. To this aim, we observe that the employment of an exact line search implies $\nabla f(x_{k+1})^T d_k = 0$, and hence the search direction of the Quasi-Newton method at iteration $k + 1$ is

$$d_{k+1} = -H_{k+1} \nabla f(x_{k+1}) = -\nabla f(x_{k+1}) + \frac{\nabla f(x_{k+1})^T y_k}{y_k^T d_k} d_k, \quad (18.13)$$

where we used (18.12) and (18.11).

The direction defined by (18.13) corresponds to the search direction generated by the Hestenes-Stiefel conjugate gradient method (note that in the case of the exact line search, the latter method is equivalent to the Polyak-Polak-Ribière conjugate gradient method).

18.4.3 Limited-Memory Quasi-Newton Methods

Limited-memory Quasi-Newton methods implicitly store the matrix H_k by using a prefixed number $m \geq 1$ of pairs $\{s_i, y_i\}$ in formulas (18.10) and (18.11). In particular, the product $H_k \nabla f(x_k)$ can be obtained by a sequence of scalar products and vector sums depending on $\nabla f(x_k)$ and on the pairs $\{s_i, y_i\}$. Once the point x_k has been updated, the “oldest” pair $\{s_i, y_i\}$ is replaced by the new pair $\{s_k, y_k\}$. In this way, the set of pairs contains information on the last m iterations. The practical experience shows that suitable values of m are in the interval $[2, 20]$.

Limited-memory Quasi-Newton methods can be viewed as an extension of memoryless Quasi-Newton methods, since they use, besides to the current pair

(s_k, y_k) , further pairs deriving from the past iterations. First we show that the product $H_k \nabla f(x_k)$ can be efficiently obtained by a recursive procedure. Before to formally define the procedure, we provide a brief justification.

Let x_k be the current point, let $\{s_i, y_i\}$, $i = k-m, \dots, k-1$ be the set of stored pairs, and let H_{k-m} be the approximation of the inverse of the Hessian used at iteration $k-m$. Starting from the matrix H_{k-m} and applying sequentially formula (18.10) we obtain

$$\begin{aligned} H_k &= [V_{k-1}^T \dots V_{k-m}^T] H_{k-m} [V_{k-m} \dots V_{k-1}] \\ &\quad + \rho_{k-m} [V_{k-1}^T \dots V_{k-m+1}^T] s_{k-m} s_{k-m}^T [V_{k-m+1} \dots V_{k-1}] \\ &\quad + \rho_{k-m+1} [V_{k-1}^T \dots V_{k-m+2}^T] s_{k-m+1} s_{k-m+1}^T [V_{k-m+2} \dots V_{k-1}] \\ &\quad + \dots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T. \end{aligned}$$

Set $q_k = \nabla f(x_k)$ and define, for $i = k-1, \dots, k-m$, the vectors

$$q_i = V_i \dots V_{k-1} \nabla f(x_k).$$

It follows

$$q_i = V_i q_{i+1} = q_{i+1} - \rho_i y_i s_i^T q_{i+1},$$

from which, setting $\alpha_i = \rho_i s_i^T q_{i+1}$, we obtain

$$q_i = q_{i+1} - \alpha_i y_i.$$

Using the vectors q_i we can write

$$\begin{aligned} H_k \nabla f(x_k) &= [V_{k-1}^T \dots V_{k-m}^T] H_{k-m} q_{k-m} \\ &\quad + \rho_{k-m} [V_{k-1}^T \dots V_{k-m+1}^T] s_{k-m} s_{k-m}^T q_{k-m+1} \\ &\quad + \rho_{k-m+1} [V_{k-1}^T \dots V_{k-m+2}^T] s_{k-m+1} s_{k-m+1}^T q_{k-m+2} \\ &\quad + \dots \\ &\quad + \rho_{k-1} s_{k-1} s_{k-1}^T q_k, \end{aligned}$$

from which, recalling the definition of the scalars α_i , it follows

$$\begin{aligned}
 H_k \nabla f(x_k) = & [V_{k-1}^T \quad \dots \quad V_{k-m+2}^T V_{k-m+1}^T V_{k-m}^T] H_{k-m} q_{k-m} \\
 & + [V_{k-1}^T \quad \dots \quad V_{k-m+1}^T] \alpha_{k-m} s_{k-m} \\
 & + [V_{k-1}^T \quad \dots \quad V_{k-m+2}^T] \alpha_{k-m+1} s_{k-m+1} \\
 & + \dots \\
 & + \alpha_{k-1} s_{k-1}.
 \end{aligned} \tag{18.14}$$

Formula (18.14) refers to the BFGS method. We observe that formula (18.14) depends on the matrix H_{k-m} defined at iteration $k - m$.

The idea underlying L-BFGS methods is that of replacing in (18.14) the “true” matrix H_{k-m} by a generic matrix H_k^0 (that we can think suitably sparse), thus obtaining

$$\begin{aligned}
 H_k \nabla f(x_k) = & [V_{k-1}^T \quad \dots \quad V_{k-m+2}^T V_{k-m+1}^T V_{k-m}^T] H_k^0 q_{k-m} \\
 & + [V_{k-1}^T \quad \dots \quad V_{k-m+1}^T] \alpha_{k-m} s_{k-m} \\
 & + [V_{k-1}^T \quad \dots \quad V_{k-m+2}^T] \alpha_{k-m+1} s_{k-m+1} \\
 & + \dots \\
 & + \alpha_{k-1} s_{k-1}.
 \end{aligned} \tag{18.15}$$

Now set $r_{k-m-1} = H_k^0 q_{k-m}$ and define the following vectors r_i for $i = k - m, \dots, k - 1$

$$r_i = V_i^T r_{i-1} + \alpha_i s_i. \tag{18.16}$$

We have

$$r_i = r_{i-1} + \rho_i y_i^T r_{i-1} s_i + \alpha_i s_i,$$

from which, setting $\beta_i = \rho_i y_i^T r_{i-1}$ we obtain

$$r_i = r_{i-1} + (\alpha_i - \beta_i) s_i.$$

Using (18.15) and the definition (18.16) of the vectors r_i it is possible to show that

$$H_k \nabla f(x_k) = r_{k-1}. \quad (18.17)$$

Indeed we have

$$\begin{aligned} r_{k-m} &= V_{k-m}^T H_k^0 q_{k-m} + \alpha_{k-m} s_{k-m} \\ r_{k-m+1} &= V_{k-m+1}^T [V_{k-m}^T H_k^0 q_{k-m} + \alpha_{k-m} s_{k-m}] + \alpha_{k-m+1} s_{k-m+1} \\ &= V_{k-m+1}^T V_{k-m}^T H_k^0 q_{k-m} + V_{k-m+1}^T \alpha_{k-m} s_{k-m} + \alpha_{k-m+1} s_{k-m+1} \\ &\vdots \\ r_{k-1} &= V_{k-1}^T \cdots V_{k-m+2}^T V_{k-m+1}^T V_{k-m}^T H_k^0 q_{k-m} \\ &\quad + V_{k-1}^T \cdots V_{k-m+2}^T V_{k-m+1}^T \alpha_{k-m} s_{k-m} \\ &\vdots \\ &\quad + \alpha_{k-1} s_{k-1}, \end{aligned}$$

so that, taking into account (18.15), it follows that (18.17) holds.

Using the above formulas we can define the procedure for computing $H_k \nabla f(x_k)$.

Procedure (HG)

```

Set  $q_k = \nabla f(x_k)$ ;
For  $i = k - 1, k - 2, \dots, k - m$ 
    set  $\alpha_i = \rho_i s_i^T q_{i+1}$ 
    set  $q_i = q_{i+1} - \alpha_i y_i$ 
End For
Set  $r_{k-m-1} = H_k^0 q_{k-m}$ 
For  $i = k - m, k - m + 1, \dots, k - 1$ 
    set  $\beta_i = \rho_i y_i^T r_{i-1}$ 
    set  $r_i = r_{i-1} + s_i(\alpha_i - \beta_i)$ 
End For
Set  $H_k \nabla f(x_k) = r_{k-1}$  and exit.

```

The limited-memory BFGS method is defined in the following scheme, where the initial point x_0 and the integer m are given.

Algorithm L-BFGS

For $k = 0, \dots$

choose H_k^0 ;

compute $d_k = -H_k \nabla f(x_k)$ using Procedure HG;

set $x_{k+1} = x_k + \alpha_k d_k$ where α_k is computed in such a way that the Wolfe conditions hold;

If $k > m$ delete the pair $\{s_{k-m}, y_{k-m}\}$ from the storage;

Compute and store s_k and y_k .

Without considering the product $H_k^0 q_{k-m}$, Procedure HG requires $4mn$ multiplications. Note that H_k^0 can be chosen without particular constraints and can be modified at each iteration. A suitable choice is that of setting $H_k^0 = \gamma_k I$, with

$$\gamma_k = \frac{(s_{k-1})^T y_{k-1}}{(y_{k-1})^T y_{k-1}}.$$

18.5 Exercises

18.1 Define a computer code of the zero memory BFGS algorithm, employing a weak Wolfe linesearch and perform some numerical experiments.

18.2 Define a computer code of a truncated Newton algorithm, employing the conjugate gradient method for approximating Newton's direction.

18.6 Notes and References

The main references for the analysis of local convergence properties of the inexact Newton methods for nonlinear equations are [34], [64] and [152]. Monotone truncated Newton methods have been defined in [189]. Truncated Newton methods, employing nonmonotone line searches and possibly generating negative curvature directions, have been presented in [124], [173] and [89]. A survey on truncated Newton methods can be found in [188]. Limited-memory quasi-Newton methods

are presented and analyzed in [166], [195]. As already observed, several works have been devoted to the construction of sparse updates in Quasi-Newton methods for nonlinear equations and unconstrained optimization, in way that the sparsity pattern of the Jacobian or the Hessian matrix (if known) is preserved. Suggested references are the works [236], [26], [242] and [246].

Chapter 19

Derivative-Free Methods for Unconstrained Optimization



In this chapter we introduce some classes of optimization methods that do not use derivatives of the objective function. After a short introduction, we consider unconstrained minimization problems and we describe some of the best known derivative-free methods. Then we study a class of globally convergent methods based on the inexact derivative-free linesearch techniques already introduced in Chap. 10. Finally, we describe some techniques employing gradient approximations and we outline the use of model-based methods. Derivative-free methods for problems with box constraints will be presented in Chap. 20. Derivative-free nonmonotone methods will be introduced in Chap. 24.

19.1 Motivation and Classification of Derivative-Free Methods

Many real applications involve optimization problems whose objective function is not analytically known. This happens, for instance, when the objective function evaluation is either the result of a numerical simulation program, typically proprietary, or it is attained by direct measurements.

From a conceptual point of view, we can imagine that a *black box* receives as input the vector of variables $x \in R^n$ and provides as output the value $f(x)$ of the function f to be minimized.

In these cases the gradient of the objective function is not available, so that *derivative-free* techniques must be used.

Derivative-free methods can be divided into three main classes:

- *finite-difference*-based methods;
- *direct search* methods;
- *model*-based methods.

In the first class finite-difference approximations are used to replace the unknown derivatives, within standard optimization methods. In particular, a first-order approximation of the i -th partial derivative can be obtained by setting

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + \epsilon e_i) - f(x)}{\epsilon}, \quad (19.1)$$

where e_i is the i -th coordinate direction and ϵ is a positive scalar *sufficiently small*.¹

However, the application of finite difference-based methods may be not suitable whenever the evaluation of the objective function is affected by noise. In general, the presence of noise leads to “bad” gradient-based directions that could be not descent directions, and this usually is reflected in line search failures.

Direct search methods are based on the strategy of sampling the objective function along suitable search directions, in order to possibly update the current solution, using the computed function values. These methods do not require finite-difference approximations of the partial derivatives, but some of these techniques can also incorporate, when possible, suitable approximations of the derivatives.

Model based methods exploit the idea of building an approximate analytical model of the objective function (revised at each major iteration, by employing the function values attained in correspondence to points suitably generated. Then the derivatives of the model can be evaluated analytically and standard optimization algorithms can be employed.

In this chapter we will refer essentially to direct search methods for unconstrained optimization problems of the form

$$\begin{aligned} & \min f(x) \\ & x \in R^n, \end{aligned}$$

where $f : R^n \rightarrow R$ is continuously differentiable.

In particular, first we will confine ourselves to consider a few well known methods originally proposed as heuristic techniques, which include the coordinate methods, the Hooke-Jeeves method, the Nelder-Mead simplex method.

Then we state convergence conditions and we define globally convergent techniques based on the use of the linesearch derivative-free algorithms introduced in Chap. 10. We show that we can globalize heuristic methods by performing periodically inexact linesearches along suitable sets of search directions.

We also give a short illustration of techniques employing approximations of derivatives, possibly in combination with direct search methods and we give a short description of model based methods. Because of space limitations, for this class of methods we will give only some indication and references to the related literature.

¹ Typically of the order $\sqrt{\eta}$, where η is the *machine precision*.

19.2 The Coordinate Method

19.2.1 Preliminary Concepts

A derivative-free method must necessarily employ at each iteration a set of search directions. Indeed, without gradient information, it is not possible to ensure that a single direction is a descent direction. The minimal requirement of a set of directions is that it contains at least a descent direction.

It can be easily verified that, given n linearly independent directions d_1, \dots, d_n , the set of $2n$ directions

$$\{d_1, \dots, d_n, -d_1, \dots, -d_n\}$$

contains at least a descent direction at x_k provided that x_k is not a stationary point ($\nabla f(x_k) \neq 0$). Indeed, we can write

$$-\nabla f(x_k) = \sum_{i=1}^n \beta_i d_i = \sum_{i \in I^+} \beta_i d_i - \sum_{i \in I^-} |\beta_i| d_i,$$

where $I^+ = \{i \in \{1, \dots, n\} : \beta_i \geq 0\}$, $I^- = \{i \in \{1, \dots, n\} : \beta_i < 0\}$. As a consequence, multiplying by $\nabla f(x_k)$ we obtain

$$-\|\nabla f(x_k)\|^2 = \sum_{i \in I^+} \beta_i \nabla f(x_k)^T d_i + \sum_{i \in I^-} |\beta_i| \nabla f(x_k)^T (-d_i) < 0,$$

from which it follows that there must exist an index i such that either $\nabla f(x_k)^T d_i < 0$ or $\nabla f(x_k)^T (-d_i) < 0$.

A set of linearly independent directions is, for instance, the set of coordinate directions $e_i, i = 1, \dots, n$:

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad e_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

In the sequel we will present coordinate methods using the coordinate directions and their opposite as the set D of search directions, that is

$$D = \{e_1, \dots, e_n, -e_1, \dots, -e_n\}. \tag{19.2}$$

From the preceding reasonings we know that, in correspondence to a non-stationary point, by sufficiently small movements along the directions of the set D , it is possible to determine a point where the function is strictly decreased.

We distinguish between methods requiring a *simple decrease* of f for updating the point and methods that perform a line search based on a condition of *sufficient decrease*, which will be considered in the sequel in a more general context.

19.2.2 Coordinate Method with Simple Decrease

The method is very simple and is based, at each iteration, on sampling the objective function at points $x_k \pm \alpha_k e_i$, $i = 1, \dots, n$, where x_k is the current point, α_k is the step-size, e_i is the i -th coordinate direction.

Figure 19.1 shows the sampling strategy of the algorithm for $n = 2$.

Each iteration terminates in one of the following two cases:

- (I) a condition of simple decrease holds for some i , that is

$$f(x_k + \alpha_k e_i) < f(x_k) \quad (\text{or } f(x_k - \alpha_k e_i) < f(x_k));$$

- (II) the function has been evaluated at $2n$ points along the $2n$ directions of the set D without obtaining a decrease, that is

$$f(x_k \pm \alpha_k e_i) \geq f(x_k) \quad i = 1, \dots, n.$$

In case (I) the new point is $x_{k+1} = x_k + \alpha_k e_i$ (or $x_{k+1} = x_k - \alpha_k e_i$), and the sampling step-size is not updated, i.e., $\alpha_{k+1} = \alpha_k$.

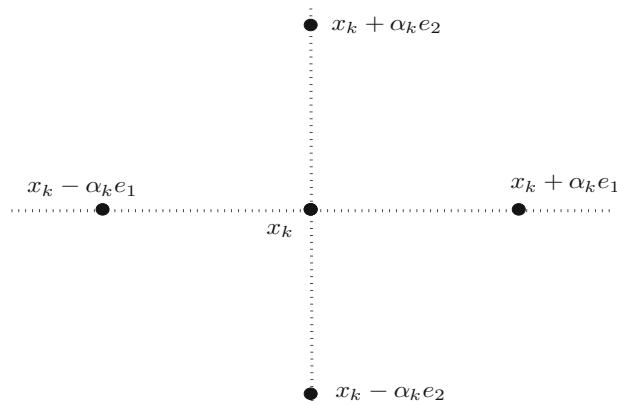


Fig. 19.1 Search along the coordinate directions

In case (II) the current point is not updated, i.e., $x_{k+1} = x_k$, and the sampling step-size is reduced by a prefixed factor, so that

$$\alpha_{k+1} = \theta\alpha_k,$$

with $\theta \in (0, 1)$.

We can formally describe the simplest conceptual version of the method.

Coordinate Method with Simple Decrease

Data: set of directions D defined by (19.2); starting point $x_0 \in R^n$, initial step-size $\alpha_0 > 0$, constant $\theta \in (0, 1)$.

For $k = 0, 1, \dots$

If there exists an index $i \in \{1, \dots, n\}$ such that

$$f(x_k + \alpha_k e_i) < f(x_k) \quad (\text{or } f(x_k - \alpha_k e_i) < f(x_k)),$$

then set

$$\begin{aligned} x_{k+1} &= x_k + \alpha_k e_i & (\text{or } x_{k+1} = x_k - \alpha_k e_i), \\ \alpha_{k+1} &= \alpha_k \end{aligned}$$

else set

$$x_{k+1} = x_k, \quad \alpha_{k+1} = \theta\alpha_k.$$

End if

End For

Remark 19.1 The method can be terminated when the step-size α_k becomes smaller than a threshold value. The result of the next proposition implies that this stopping criterion is satisfied in a finite number of iterations. \square

Remark 19.2 Note that in Algorithm 19.2.2 is completely unspecified the criterion followed for searching, at a given k , an index i (if it exists) that yields a strict decrease of f . The most common criterion could be that of considering in sequence the n vectors e_i , starting with $i = 1$ for $k = 0$. In this case, if the k -th iteration terminates in correspondence to some i_k , then at the next iteration $k + 1$ we can start by considering initially the index $i = i_k + 1$ if $i_k < n$ or $i = 1$ if $i_k = n$. But any other criterion would in principle work. \square

In order to establish a convergence result, we preliminarily state the following lemma, which follows from the properties of a compact set $S \subset R^n$.

Lemma 19.1 Let $I = \{x_1, x_2, \dots\}$ be a set of points belonging to a compact set $S \subset \mathbb{R}^n$. Assume that

$$\|x_i - x_j\| \geq \sigma > 0, \quad \text{for all } x_i, x_j \in I, x_i \neq x_j. \quad (19.3)$$

Then, the set I must have a finite number of elements.

Proof For every $x \in S$ consider the open ball $B(x; \sigma/2)$ with center x and radius $\sigma/2$. The family of these balls is obviously an open covering of S . As S is compact, we know that there must exist a finite sub-covering, that is a finite number r of balls $B(\bar{x}_i; \sigma/2)$, with $\bar{x}_i \in S$, for $i = 1, \dots, r$ such that

$$S \subset \bigcup_{i=1, \dots, r} B(\bar{x}_i; \sigma/2).$$

Given two points y, z belonging to one of these balls, say $B(\bar{x}_i; \sigma/2)$, we have

$$\|y - z\| = \|y - \bar{x}_i + \bar{x}_i - z\| \leq \|y - \bar{x}_i\| + \|\bar{x}_i - z\| < \sigma/2 + \sigma/2 = \sigma.$$

Therefore, by (19.3) every ball $B(\bar{x}_i; \sigma/2)$ can not contain two distinct points x_i, x_j of the set I . As distinct points of I belong to distinct balls, we obtain that $|I| \leq r < \infty$. \square

Now we can state the following proposition.

Proposition 19.1 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and assume that the level set \mathcal{L}_0 is compact. Let $\{\alpha_k\}$ be the sequence of scalars produced by Algorithm 19.2.2. Then we have

$$\lim_{k \rightarrow \infty} \alpha_k = 0 \quad (19.4)$$

and hence $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$.

Proof First we observe that the instructions of the algorithm imply $f(x_{k+1}) \leq f(x_k)$ and thus we have $x_k \in \mathcal{L}_0$ for all $k \geq 0$. We also have $\alpha_{k+1} \leq \alpha_k$, so that the non increasing sequence $\{\alpha_k\}$ converges to a value $\bar{\alpha} \geq 0$.

By contradiction, let us assume that the thesis is false and hence that $\bar{\alpha} > 0$. In this case there can not exist an infinite subset $K = \{k_1, k_2, \dots, k_j, \dots\}$ where the

step-size α_{k_j} is diminished by taking

$$\alpha_{k_{j+1}} = \theta \alpha_{k_j} \quad \text{for all } k_j \in K.$$

Otherwise, we could write $\alpha_{k_{j+1}} = \theta^j \alpha_0$, and this would imply, being $\theta \in (0, 1)$, that $\bar{\alpha} = 0$. Then, for k sufficiently large (say $k \geq \bar{k}$) we necessarily have

$$\alpha_k = \bar{\alpha} > 0. \quad (19.5)$$

Because of the instructions of the algorithm, this implies that the subsequence $I^* = \{x_k\}_{k \geq \bar{k}}$ is infinite and that we have $f(x_{k+1}) < f(x_k)$, at each $x_k \in I^*$, so that all points in I^* must be distinct.

We show that this yields a contradiction to the assumption $\bar{\alpha} > 0$. Let us consider any pair of points (necessarily distinct) $x_{\bar{k}+i}, x_{\bar{k}+j}$ ($i < j$) of the subsequence $\{x_k\}_{k \geq \bar{k}}$. As the points are generated by movements along directions parallel to the coordinate directions we can write

$$x_{\bar{k}+j} - x_{\bar{k}+i} = \bar{\alpha} \sum_{t=\bar{k}+i}^{\bar{k}+j-1} d_t, \quad (19.6)$$

where $d_t \in \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\}$. Since $x_{\bar{k}+i}$ and $x_{\bar{k}+j}$ are distinct we obtain

$$x_{\bar{k}+j} - x_{\bar{k}+i} = \bar{\alpha} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \neq 0, \quad (19.7)$$

where a_1, a_2, \dots, a_n are integers not all zero. From (19.7) we get

$$\|x_{\bar{k}+i} - x_{\bar{k}+j}\| \geq \bar{\alpha}, \quad (19.8)$$

for all i, j such that $i < j$. But then, as all points belong to the compact set \mathcal{L}_0 , Lemma 19.1 would imply that I^* has a finite number of elements and this yields a contradiction to our assumptions on I^* .

It can be concluded that (19.4) must hold. As $\|\pm e_i\| = 1$ for all i , this implies $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$. \square

We can state the global convergence result.

Proposition 19.2 Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be continuously differentiable over \mathbb{R}^n and assume that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence generated by the coordinate method (Algorithm 19.2.2). Then $\{x_k\}$ admits limit points and there exists at least a limit point, which is a stationary point.

Proof We have $f(x_{k+1}) \leq f(x_k)$ for all $k \geq 0$, and hence the points of the sequence $\{x_k\}$ belong to the compact set \mathcal{L}_0 . This implies that $\{x_k\}$ admits limit points. Proposition 19.1 and the instructions of the algorithm imply that there exists an infinite subset $K_1 \subseteq \{0, 1, \dots\}$ such that, for all $k \in K_1$ we have

$$\alpha_{k+1} = \theta \alpha_k.$$

Let us consider the subsequence $\{x_k\}_{K_1}$: as the points of $\{x_k\}_{K_1}$ belong to the compact set \mathcal{L}_0 , we can state that there exists a subset $K_2 \subseteq K_1$ such that

$$\lim_{k \in K_2, k \rightarrow \infty} x_k = \bar{x}. \quad (19.9)$$

On the other hand, for $k \in K_2$ and for $i = 1, \dots, n$ we must have

$$f(x_k + \alpha_k e_i) \geq f(x_k), \quad (19.10)$$

$$f(x_k - \alpha_k e_i) \geq f(x_k). \quad (19.11)$$

For $i \in \{1, \dots, n\}$, by the mean value theorem, we can write

$$f(x_k + \alpha_k e_i) = f(x_k) + \alpha_k \nabla f(u_k^i)^T e_i, \quad (19.12)$$

$$f(x_k - \alpha_k e_i) = f(x_k) - \alpha_k \nabla f(v_k^i)^T e_i, \quad (19.13)$$

where $u_k^i = x_k + \xi_k^i \alpha_k e_i$, with $\xi_k^i \in (0, 1)$, $v_k^i = x_k - \mu_k^i \alpha_k e_i$, with $\mu_k^i \in (0, 1)$.

Using (19.4) and (19.9) we obtain for $i = 1, \dots, n$

$$\lim_{k \in K_2, k \rightarrow \infty} u_k^i = \lim_{k \in K_2, k \rightarrow \infty} v_k^i = \bar{x}. \quad (19.14)$$

Substituting (19.12) in (19.10) and (19.13) in (19.11) it follows

$$\alpha_k \nabla f(u_k^i)^T e_i \geq 0, \quad (19.15)$$

$$-\alpha_k \nabla f(v_k^i)^T e_i \geq 0. \quad (19.16)$$

From (19.15) and (19.16), taking into account (19.14) and the continuity of ∇f , for $i = 1, \dots, n$ we have

$$\lim_{k \in K_2, k \rightarrow \infty} \nabla f(u_k^i)^T e_i = \nabla f(\bar{x})^T e_i \geq 0, \quad (19.17)$$

$$\lim_{k \in K_2, k \rightarrow \infty} \nabla f(v_k^i)^T e_i = \nabla f(\bar{x})^T e_i \leq 0, \quad (19.18)$$

which imply

$$\nabla f(\bar{x})^T e_i = \frac{\partial f(\bar{x})}{\partial x_i} = 0, \quad i = 1, \dots, n.$$

□

Remark 19.3 We observe that the global convergence result of the algorithm is obtained without requiring a *sufficient decrease* of the objective function. However, the algorithm is based on a suitable control mechanism of the steplength that is reduced only when a failure occurs along all the search directions. This yields local information on the objective function in a neighborhood of the failure point allowing to ensure convergence properties of the generated sequence. □

Remark 19.4 A stronger convergence result can be obtained by modifying the simple version of the coordinate method. The modification concerns the *successful* iterates, that is the iterates where a strict decrease of f has been obtained. In the presented method the updating of x_k ($x_{k+1} \neq x_k$) is performed whenever a strict decrease of f has been attained along any search direction. In this case the new point can be any of the $2n$ candidates

$$x_k \pm \alpha_k e_i \quad i = 1, \dots, n,$$

with a lower value of f with respect to $f(x_k)$.

In the modified version of the method, once all the $2n$ candidates have been evaluated, the “best” point in terms of lowest corresponding function value must be chosen as new point. Then, each iteration requires $2n$ function evaluations.

For the modified version of the method it is possible to prove a stronger convergence result. In particular, under the same assumptions of Proposition 19.2, we have that $\nabla f(x_k) \rightarrow 0$ for $k \rightarrow \infty$, and hence that *each limit point* of $\{x_k\}$ is a stationary point of f . □

An interesting variation of the coordinate method is *Rosenbrock method*, where, starting from the coordinate directions a new set of orthogonal directions is obtained at each major step by rotating the preceding set in a way that at least the initial search direction of the new set is more closely conformed to the local behavior of the function.

As we will see in the sequel, a different approach to coordinate method and to Rosenbrock method will be that of performing line searches along the coordinate axes or along the set of orthogonal directions obtained by rotation.

19.3 The Hooke-Jeeves Method

The Hooke-Jeeves method can be viewed as a variant of the coordinate method. It can be described as a sequences of major iterations, each (possibly) structured in two phases. In the first phase, called *exploratory move*, starting from the current point x_k , we perform a search along the $2n$ directions $\pm e_i, i = 1, \dots, n$ with a fixed step-size $\alpha_k > 0$. We say that this search is *successful* if, at the end of the search, we obtain a point x_{k+1} such that $f(x_{k+1}) < f(x_k)$. Note that in a successful exploratory move the movements are performed along all the coordinate directions and hence *at least one* component of the current point x_k has been modified (but we may have a greater number of modified components). If the exploratory move fails, then we proceed as in the coordinate method, that is we set $x_{k+1} = x_k$ and $\alpha_{k+1} = \theta \alpha_k$ with $\theta \in (0, 1)$.

In case of a successful exploratory move, the peculiarity of the Hooke-Jeeves method is the subsequent *pattern search*, or *pattern move*, that is the attempt of exploiting also the direction $(x_{k+1} - x_k)$ (which could be a “good” direction having produced a decrease of f) for generating a point \hat{x}_{k+1} that is “temporarily” accepted. In particular we have

$$\hat{x}_{k+1} = x_{k+1} + (x_{k+1} - x_k),$$

and this point is temporarily accepted even if

$$f(\hat{x}_{k+1}) \geq f(x_{k+1}).$$

Starting from \hat{x}_{k+1} a new search along the coordinate directions is performed:

- if this search “fails”, i.e., $f(\hat{x}_{k+1} \pm \alpha_k e_i) \geq f(x_{k+1})$ for $i = 1, \dots, n$, then the point \hat{x}_{k+1} is discarded and the new iteration starts from x_{k+1} ;
- if the search “is a successful search”, i.e., if for some i we have

$$f(\hat{x}_{k+1} + \alpha_k e_i) < f(x_{k+1}) \quad (\text{or } f(\hat{x}_{k+1} - \alpha_k e_i) < f(x_{k+1})),$$

then we set

$$x_{k+2} = \hat{x}_{k+1} + \alpha_k e_i \quad (\text{or } x_{k+2} = \hat{x}_{k+1} - \alpha_k e_i).$$

Fig. 19.2 Points generated by the Hooke-Jeeves method

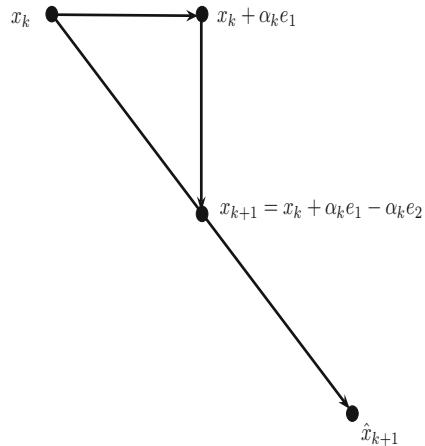


Figure 19.2 shows the strategy of the method. In the examples shown in the figure we suppose that

$$f(x_k + \alpha_k e_1 - \alpha_k e_2) < f(x_k + \alpha_k e_1) < f(x_k).$$

As regards the global convergence properties, we observe that all the points generated by the algorithm (including those temporarily accepted) are obtained by movements with assigned amplitude along directions parallel to the coordinate directions.

Indeed we have

$$x_{k+1} = x_k + \sum_{j=1}^{2n} \alpha_k^j e_j$$

$$\hat{x}_{k+1} = x_{k+1} + \sum_{j=1}^{2n} \alpha_k^j e_j,$$

where either $\alpha_k^j = \alpha_k$ or $\alpha_k^j = 0$. Therefore we can fully repeat the reasonings used in the proofs of Propositions 19.1 and 19.2. As a consequence, if $\{x_k\}$ is the sequence generated by the Hooke-Jeeves method, and \mathcal{L}_0 is compact, we can state that there exists at least a limit point of $\{x_k\}$ which is a stationary point of f .

19.4 The Nelder-Mead Method

The Nelder-Mead method is one of the most popular and used direct search method. It is also known as (*Nelder-Mead simplex method*) for the reasons explained below. Note that there is no relationship with the simplex method for linear programming. The name “Nelder-Mead simplex” follows from the fact that the method uses at each iteration the simplex of $n + 1$ points, that is the *convex hull* of such a points.

Given a simplex S with vertices x_1, \dots, x_{n+1} , we denote by $V(S)$ the following $n \times n$ matrix of *simplex directions*

$$V(S) = [x_2 - x_1, x_3 - x_1, \dots, x_{n+1} - x_1].$$

If the matrix $V(S)$ is nonsingular then the simplex S is said *nonsingular*. The *diameter* $\text{diam}(S)$ of the simplex is defined as follows

$$\text{diam}(S) = \max_{1 \leq i, j \leq n+1} \|x_i - x_j\|.$$

The Nelder-Mead method uses at each iteration a simplex S with vertices x_1, \dots, x_{n+1} , ordered on the basis of the corresponding values of the objective function in such a way that we have

$$f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1}).$$

The point x_1 represents the *best vertex*, the point x_{n+1} represents the *worse vertex*.

We denote by \bar{x} the centroid of the best n points, that is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

The strategy of the method is that of trying to replace the worse vertex x_{n+1} by a point obtained by taking

$$x = \bar{x} + \mu(\bar{x} - x_{n+1}) \quad \mu \in R.$$

The value of μ is chosen in a finite set and defines the kind of iteration.

Typical values of μ are:

- $\mu = 1$, which indicates a *reflection*;
- $\mu = 2$, which indicates an *expansion*;
- $\mu = 1/2$, which indicates an *outside contraction*;
- $\mu = -1/2$, which indicates an *inside contraction*.

The choice of the trial points that could replace the worse vertex depends on the value of the objective function attained by the reflection and in some vertices of the simplex. In some cases the method may perform a *shrink* of the simplex. In

particular, all the vertices except for the best vertex x_1 are replaced. The new vertices are defined by taking

$$x_1 + \gamma(x_i - x_1) \quad i = 2, \dots, n+1,$$

with $\gamma \in (0, 1)$ (the typical value is 1/2).

Now we can formally define the Nelder-Mead method.

The Nelder-Mead Method

Data: set of starting points $X = [x_1, \dots, x_{n+1}]$; numbers $\gamma, \mu^{ci}, \mu^{co}, \mu^r, \mu^e$ such that $0 < \gamma < 1 - 1 < \mu^{ci} < 0 < \mu^{co} < \mu^r < \mu^e$.

For $k = 0, 1, \dots$

Step 1: sorting. Sort the vertices in such a way that

$$f(x_1) \leq f(x_2) \leq \dots f(x_{n+1}).$$

Step 2: reflection. Set $x^r = \bar{x} + \mu^r(\bar{x} - x^{n+1})$. If $f(x_1) \leq f(x^r) < f(x_{n+1})$, update X replacing x_{n+1} by x^r and terminate the iteration, provided that $f(x^r) < f(x_n)$.

Step 3: expansion. If $f(x^r) < f(x_1)$ determine the expansion point $x^e = \bar{x} + \mu^e(\bar{x} - x^{n+1})$. If $f(x^e) < f(x^r)$ update X replacing x_{n+1} by x^e and terminates the iteration. Otherwise, update X replacing x_{n+1} by x^r and terminate the iteration.

Step 4: contraction. If $f(x^r) \geq f(x_n)$ then

- (a) *outside contraction:* if $f(x^r) < f(x_{n+1})$ then set $x^{ce} = \bar{x} + \mu^{co}(\bar{x} - x^{n+1})$. If $f(x^{ce}) \leq f(x^r)$ then update X replacing x_{n+1} by x^{ce} , and terminate the iteration.
- (b) *inside contraction:* if $f(x^r) \geq f(x_{n+1})$ then set $x^{ci} = \bar{x} + \mu^{ci}(\bar{x} - x^{n+1})$. If $f(x^{ci}) < f(x_{n+1})$ then update X replacing x_{n+1} by x^{ci} , and terminate the iteration.

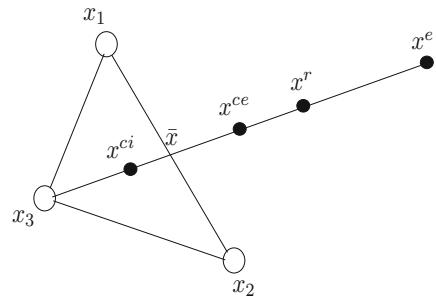
Step 5: shrink. Compute the function value at the n points defined by $x_1 + \gamma(x_i - x_1)$ for $i = 2, \dots, n+1$, update X replacing the points x_2, \dots, x_{n+1} with the generated points, and terminate the iteration.

End For

Figure 19.3 shows the points generated by the method in the case of $n = 2$. We observe that the method requires a number of function evaluations equal to:

- 1 if the iteration is a *reflection*;
- 2 if the iteration is either an *expansion* or a *contraction*;
- $n + 2$ if the iteration is a *shrink*.

Fig. 19.3 Points generated by the Nelder-Mead method



A reasonable stopping criterion can be that of terminating the iterations whenever the diameter of the simplex becomes smaller than a prefixed threshold value (whose typical value is 10^{-5}).

The method is globally convergent in the one dimensional case. For $n \geq 2$ the convergence of the method can not be ensured. Indeed, there exist counter-examples with dimension 2 and continuously differentiable function where the method fails, that is generates a sequence convergent towards a non stationary point. However, even if the method does not present global convergence properties, it is computationally efficient.

It can be shown that if Step 5 is never executed then the average value

$$\frac{1}{n+1} \sum_{i=1}^{n+1} f(x_i)$$

decreases at each iteration. A sufficient condition to ensure that Step 5 is never executed is that the function f is *strictly convex*.

Different modified versions of the Nelder-Mead method with global convergence properties have been proposed in the literature.

19.5 Linesearch-Based Methods

In this section we consider derivative-free techniques based on the use of the derivative-free linesearch techniques introduced in Chap. 10. The algorithms we will consider for minimizing an unconstrained continuously differentiable function $f : R^n \rightarrow R$ generate a sequence of points $\{x_k\}$ in R^n such that at each k we perform a linesearch along a search direction d_k . As we know from Chap. 10, under appropriate assumptions, we can construct derivative-free algorithms such that, given $d_k \in R^n$, $d_k \neq 0$ we have

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

where $\alpha_k \in R$ (possibly $\alpha_k = 0$).

In particular, as derivative-free line search, we refer to the algorithm of Armijo-Goldstein-type, introduced in Chap. 10 and here reported to help the reader, which employs bidirectional searches and admits step expansions.

Derivative-Free Armijo-Goldstein-Type Linesearch

Data. $\Delta_k > 0, \gamma_2 > \gamma_1 > 0, \delta \in (0, 1), \rho_k > 0$.

1. Set $\alpha = \Delta_k$
2. **While** $f(x_k \pm \alpha d_k) > f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$ **do**
 - If** $\alpha \|d_k\| < \rho_k$ **then**
 - set $\eta_k = \alpha, \alpha_k = 0$ and **exit**.
 - Else**
 - set $\alpha = \delta \alpha$.
 - End If****End while**

3. Let $u \in \{-1, 1\}$ be such that

$$f(x_k + u \alpha d_k) \leq f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$$

and set $\alpha = u \alpha$.

4. If $|\alpha| < \Delta_k$ set $\alpha_k = \alpha$ and **exit**.
5. **While**

$$f(x_k + \alpha d_k) < f(x_k) - \gamma_2 \alpha^2 \|d_k\|^2,$$

$$f(x_k + (\alpha/\delta) d_k) < \min \left\{ f(x_k + \alpha d_k), f(x_k) - \gamma_1 (\alpha/\delta)^2 \|d_k\|^2 \right\}$$

set $\alpha = \alpha/\delta$.

- End while**
6. Set $\alpha_k = \alpha$ and **exit**. □

We have shown (see Proposition 10.7) also that, assuming compact the level set \mathcal{L}_0 , we can guarantee that the linesearch algorithm satisfies

- (a) $f(x_{k+1}) \leq f(x_k)$;
- (b) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c) $\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$.

In the method presented here the search directions are sequentially selected in a given set D . The assumptions on this set are considered in the next paragraph.

19.5.1 Basic Assumptions and Convergence Conditions

Suppose first that the sequence of search directions $\{d_k\}$ satisfies the following assumption, which requires, in essence, that ultimately a set of n uniformly linearly independent search directions is employed cyclically.

Assumption 19.1 *There exist an integer $N \geq n > 0$ and integer sequences $\{j_k^i\}$, for $i = 1, \dots, n$, such that*

- (a) $k \leq j_k^1 \leq j_k^2 \leq \dots \leq j_k^n \leq k + N$ for each $k \geq 0$;
- (b) *every limit point of the sequence of $n \times n$ matrices*

$$P_k = \left(\frac{d_{j_k^1}}{\|d_{j_k^1}\|} \frac{d_{j_k^2}}{\|d_{j_k^2}\|} \dots \frac{d_{j_k^n}}{\|d_{j_k^n}\|} \right) \quad (k = 0, 1, \dots),$$

where $\|d_{j_k^i}\| > 0$ for all $k \geq 0$ and $i = 1, \dots, n$, is a non singular matrix in $R^{n \times n}$. \square

Remark 19.5 It is easily seen that the assumption given above is satisfied in a scheme where the coordinate directions (or a set of linear independent directions) are employed cyclically. However the assumption is satisfied also when a periodic search along the n linearly independent directions is performed only every N steps, for globalizing any technique such that $f(x_{k+1}) \leq f(x_k)$. \square

We can now establish the following result.

Proposition 19.3 *Let $f : R^n \rightarrow R$ be a continuously differentiable function and assume that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence of points produced by an algorithm of the form $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ for all k and $\alpha_k \in R$. Suppose that:*

- (i) *Assumption 19.1, is satisfied;*
- (ii) *the step-size α_k along d_k is computed using Algorithm 19.5;*
- (iii) *we have $\rho_k \rightarrow 0$ for $k \rightarrow \infty$.*

Then the algorithm produces an infinite sequence that admits limit points and every limit point \bar{x} of $\{x_k\}$ is in \mathcal{L}_0 and satisfies $\nabla f(\bar{x}) = 0$.

Proof Taking into account the assumptions on f and \mathcal{L}_0 and assumption (ii), we have that all the hypotheses of Proposition 10.7 are satisfied and hence the assertions of this proposition must hold. It follows, in particular, that $\{f(x_k)\}$ converges, that $x_k \in \mathcal{L}_0$ for all k , that $\{x_k\}$ has limit points, that every limit point of the sequence is

in \mathcal{L}_0 and that

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (19.19)$$

We must show that every limit point of $\{x_k\}$ is also a stationary point of f .

Let $\bar{x} \in \mathcal{L}_0$ be a limit point of $\{x_k\}$ and let $\{x_k\}_K$ be a subsequence converging to \bar{x} . Consider the search directions $d_{j_k^i}$, for $i = 1, \dots, n$, introduced in Assumption 19.1 and let $p_k^i = d_{j_k^i} / \|d_{j_k^i}\|$, for $i = 1, \dots, n$ be the columns of the matrix P_k defined there. As all the sequences $\{p_k^i\}$ are bounded there exists a subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \rightarrow \infty} p_k^i = \bar{p}^i, \quad i = 1, \dots, n. \quad (19.20)$$

By Assumption 19.1, we have that the vectors \bar{p}^i , $i = 1, \dots, n$ are linearly independent. By (19.19), as $k \leq j_k^i \leq k + N$, it can be easily established, by induction, that all the points $x_{j_k^i}$ converge to \bar{x} for $k \in K_1, k \rightarrow \infty$ and for all $i = 1, \dots, n$. By Proposition 10.7, we have that:

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_{j_k^i})^T d_{j_k^i}}{\|d_{j_k^i}\|} = \nabla f(\bar{x})^T \bar{p}^i = 0, \quad i = 1, \dots, n. \quad (19.21)$$

Since vectors \bar{p}^i are linearly independent, we obtain $\nabla f(\bar{x}) = 0$. \square

We note that, as Algorithm 19.5 makes use of bidirectional searches, we must explore the $2n$ directions $\pm d_1, \pm d_2, \dots, \pm d_n$. However, we can define globally convergent algorithms using a number of directions less than $2n$. To this aim we introduce concepts related to the *conical combination* of vectors.

Given the vectors $d_i \in R^n$, $i = 1, \dots, r$ and a set $S \subseteq R^n$, we say that $S \subseteq \text{cone}\{d_1, \dots, d_r\}$ if each element of S can be expressed as conical combination of the vectors d_1, \dots, d_r , that is, if for all $x \in S$, there exist $\alpha_i \geq 0$, $i = 1, \dots, r$ such that $x = \sum_{i=1}^r \alpha_i d_i$. The property of the set of $2n$ coordinate directions (fundamental in the context of derivative-free methods) is that

$$R^n = \text{cone}\{e_1, \dots, e_n, -e_1, \dots, -e_n\},$$

that is, each vector of R^n can be expressed as *conical combination* of these directions. Indeed, for each $x \in R^n$ we can write

$$x = \sum_{i=1}^n x_i e_i = \sum_{i \in I^+} x_i e_i + \sum_{i \in I^-} |x_i|(-e_i),$$

where $I^+ = \{i : x_i \geq 0\}$, $I^- = \{i : x_i < 0\}$.

Then the property required to a set $D = \{d_1, \dots, d_r\}$ of search directions, for a derivative-free method, is

$$R^n = \text{cone}(D), \quad (19.22)$$

i.e., the columns of D positively span R^n . In particular, it can be easily verified (by the same reasonings used at the beginning of Sect. 19.2) that given a point $x_k \in R^n$ such that $\nabla f(x_k) \neq 0$, there exists a direction $d_i \in D$ which is a descent direction for f at x_k . Note that setting

$$D = \left\{ e_1, e_2, \dots, e_n, -\sum_{i=1}^n e_i \right\} \quad (19.23)$$

condition (19.22) holds. Now we will show that a set D satisfying property (19.22) contains at least $n + 1$ directions.

Proposition 19.4 *Let $D = \{d_1, \dots, d_r\}$ a set of vectors in R^n and assume that (19.22) holds. Then $r \geq n + 1$.*

Proof Let $d_h \neq 0$ with $h \in \{1, \dots, r\}$. If (19.22) holds then d_h and $-d_h$ can be expressed as conical combination of d_1, \dots, d_r , that is:

$$\begin{aligned} d_h &= \sum_{i=1}^r \alpha_i d_i, \quad \alpha_i \geq 0, \quad \sum_{i=1}^r \alpha_i > 0, \\ -d_h &= \sum_{i=1}^r \sigma_i d_i, \quad \sigma_i \geq 0, \quad \sum_{i=1}^r \sigma_i > 0, \end{aligned}$$

whence it follows (summing and assuming, without loss of generality, $\alpha_1 + \sigma_1 > 0$)

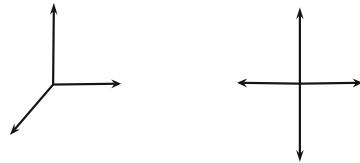
$$d_1 = \sum_{i=2}^r \tilde{\alpha}_i d_i, \quad \tilde{\alpha}_i = -\frac{\alpha_i + \sigma_i}{\alpha_1 + \sigma_1}. \quad (19.24)$$

For every $x \in R^n$, using (19.22) and (19.24), we can write

$$x = \sum_{i=1}^r \gamma_i d_i = \sum_{i=2}^r \beta_i d_i,$$

where $\beta_i = \gamma_1 \tilde{\alpha}_i + \gamma_i$, $i = 2, \dots, r$. Then any vector $x \in R^n$ can be expressed as linear combination of the vectors d_2, \dots, d_r . Therefore, the set $\{d_2, \dots, d_r\}$ must

Fig. 19.4 A minimal positive basis and a maximal positive basis in R^2



contain a basis of R^n , and hence its cardinality is greater or equal to n , and this implies $r - 1 \geq n$. \square

We say that a set $D = \{d_1, \dots, d_r\}$ is *positively dependent* if a vector of D can be expressed as conic combination of the other vectors.

Otherwise the set D is said *positively independent*. A *positive basis* is a positively independent set $D = \{d_1, \dots, d_r\}$ such that (19.22) holds. It can be easily verified that, given n linearly independent vectors d_1, \dots, d_n , the sets

$$\{d_1, \dots, d_n, -d_1, \dots, -d_n\}, \quad \left\{ d_1, d_2, \dots, d_n, -\sum_{i=1}^n d_i \right\}.$$

are positive bases.

Proposition 19.4 states that a positive basis contains at least $n + 1$ elements. It can be proved that a positive basis contains at most $2n$ vectors. A positive basis with $n + 1$ elements is said *minimal*, a positive basis with $2n$ elements is said *maximal*.

Examples of minimal and maximal positive bases in R^2 are shown in Fig. 19.4.

Now, in order to establish a convergence results for linesearch based methods employing positive bases, we will assume that the sequence $\{d_k\}$ satisfies the following conditions.

Assumption 19.2 There exist integers $N > 0$ and $r > 0$ and integer sequences $\{j_k^i\}$, $(i = 1, \dots, r)$, such that

- (a) $k \leq j_k^1 \leq j_k^2 \leq \dots \leq j_k^r \leq k + N$ for each $k \geq 0$;
- (b) every limit point of the sequence of $n \times r$ matrices

$$Q_k = \begin{pmatrix} \frac{d_{j_k^1}}{\|d_{j_k^1}\|} & \frac{d_{j_k^2}}{\|d_{j_k^2}\|} & \cdots & \frac{d_{j_k^r}}{\|d_{j_k^r}\|} \end{pmatrix} \quad (k = 0, 1, \dots),$$

where $\|d_{j_k^i}\| > 0$ for all $k \geq 0$ and $i = 1, \dots, r$, is a matrix in $R^{n \times r}$ whose columns positively span R^n . \square

It can be easily verified that the positive bases considered above (where $r = 2n$ or $r = n + 1$) satisfy the conditions of Assumption 19.2.

In this case, we suppose that the line search is carried out only for $\alpha \geq 0$ along each direction, and hence we refer to the linesearch algorithm defined in Chap. 10 and here reported again.

Derivative-Free Linesearch with Positive Steps

Data. $\Delta_k > 0, \gamma_2 > \gamma_1 > 0, \delta \in (0, 1), \rho_k > 0$.

1. Set $\alpha = \Delta_k$
2. **While** $f(x_k + \alpha d_k) > f(x_k) - \gamma_1 \alpha^2 \|d_k\|^2$ **do**
 - If $\alpha \|d_k\| < \rho_k$ **then**
 - set $\eta_k = \alpha, \alpha_k = 0$ and **exit**.
 - Else**
 - set $\alpha = \delta \alpha$.
 - End If****End while**
3. If $\alpha < \Delta_k$ set $\alpha_k = \alpha$ and **exit**.
4. **While**

$$f(x_k + \alpha d_k) < f(x_k) - \gamma_2 \alpha^2 \|d_k\|^2,$$

$$f(x_k + (\alpha/\delta)d_k) < \min \left\{ f(x_k + \alpha d_k), f(x_k) - \gamma_1(\alpha/\delta)^2 \|d_k\|^2 \right\}$$

$$\text{set } \alpha = \alpha/\delta.$$

End while

5. Set $\alpha_k = \alpha$ and **exit**. □

In this case, assuming compact the level set \mathcal{L}_0 and recalling Proposition 10.8, we can guarantee that the linesearch algorithm satisfies the properties

- (a) $f(x_{k+1}) \leq f(x_k)$;
- (b) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c) $\liminf_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| \geq 0$.

Then we can state a convergence result similar to that given in Proposition 19.3.

Proposition 19.5 *Let $f : R^n \rightarrow R$ be a continuously differentiable function and assume that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence of points produced by an algorithm of the form $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ for all k and $\alpha_k \in R, \alpha \geq 0$. Suppose that:*

- (i) *Assumption 19.2 is satisfied;*
- (ii) *the step-size α_k along d_k is computed using Algorithm 19.5.1;*
- (iii) *we have $\rho_k \rightarrow 0$ for $k \rightarrow \infty$*

Then the algorithm produces an infinite sequence that admits limit points and every limit point \bar{x} of $\{x_k\}$ is in \mathcal{L}_0 and satisfies $\nabla f(\bar{x}) = 0$.

Proof Reasoning as in the proof of Proposition 19.3, we can establish that $\{f(x_k)\}$ converges, that $x_k \in \mathcal{L}_0$ for all k , that $\{x_k\}$ has limit points, that every limit point of the sequence is in \mathcal{L}_0 and that

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (19.25)$$

We must show that every limit point of $\{x_k\}$ is also a stationary point of f .

Let \bar{x} be a limit point of $\{x_k\}$ and denote by $\{x_k\}_K$ a subsequence converging to $\bar{x} \in \mathcal{L}_0$. Consider the search directions introduced in Assumption 19.2

$$d_{j_k^i}, \quad j_k^i \in \{k, k+1, \dots, k+N\} \quad i = 1, \dots, r,$$

and let q_k^i be the columns of the matrix Q_k defined there. As all the sequences $\{q_k^i\}$ are bounded there exists a subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \rightarrow \infty} q_k^i = \bar{q}^i, \quad i = 1, \dots, r. \quad (19.26)$$

By Assumption 19.2, we have that \bar{q}^i , $i = 1, \dots, r$ represent a positive basis in R^n . By (19.25) we have that points $x_{j_k^i}$ converge to \bar{x} for $k \in K_1, k \rightarrow \infty$ and for all $i = 1, \dots, r$. As α_k is computed through Algorithm 19.5.1 we obtain, as already remarked in condition (c) stated before the proposition, that:

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_{j_k^i})^T d_{j_k^i}}{\|d_{j_k^i}\|} = \nabla f(\bar{x})^T \bar{q}^i \geq 0, \quad i = 1, \dots, r. \quad (19.27)$$

As vectors \bar{q}^i form a positive basis in R^n , we can write $-\nabla f(\bar{x}) = \sum_{i=1}^r \zeta_i \bar{q}^i$, $\zeta_i \geq 0$, so that, by (19.27), we have $-\|\nabla f(\bar{x})\|^2 = \sum_{i=1}^r \zeta_i \nabla f(\bar{x})^T \bar{q}^i \geq 0$ and thus, we obtain $\nabla f(\bar{x}) = 0$. \square

19.5.2 Globalization of Direct Search Methods Through Line Searches

We consider here applications of the derivative-free linesearch algorithms introduced in the preceding sections to the globalization of some direct search methods. In particular, we will define by means of a single scheme, a linesearch-based versions of the coordinate method and of the Hooke-Jeeves method.

In the following scheme the coordinate method can be obtained by deleting Step 2. We suppose also that the linesearches are carried out with the bidirectional search defined in Algorithm 19.5, which will be referred to as Algorithm LS($d_k, \Delta_k, \rho_k, \alpha_k$) for indicating input and output parameters.

Coordinate and Hooke-Jeeves Method with Linesearches

Data. Starting point $x_0 \in R^n$, $\theta \in (0, 1)$, $\rho_0 > 0$, $D = \{e_1, e_2, \dots, e_n\}$.

Set $k = 0$.

For $\ell = 0, 1, \dots$

 Set $y^0 = x_k$.

Step 1. *Coordinate search, (exploratory move)*

For $i = 1, \dots, n$

 set $d_k = e_i$;

 choose an initial step-size $\Delta_k > 0$ and calculate step α_k along d_k using Algorithm LS($d_k, \Delta_k, \rho_k, \alpha_k$);

 set $x_{k+1} = x_k + \alpha_k d_k$;

 if $\alpha_k = 0$, set $\rho_{k+1} = \theta \rho_k$, otherwise set $\rho_{k+1} = \rho_k$

 set $k = k + 1$.

End For

Step 2. *Pattern move*

 Set $d_k = x_k - y^0$.

 Choose an initial step-size $\Delta_k > 0$ and calculate step α_k along d_k using Algorithm LS($d_k, \Delta_k, \rho_k, \alpha_k$);

 set $x_{k+1} = x_k + \alpha_k d_k$;

 set $k = k + 1$.

End For

The convergence of the algorithm is established in the next proposition.

Proposition 19.6 *Let $f : R^n \rightarrow R$ be a continuously differentiable function and assume that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence of points produced by Algorithm 19.5.2 (where Step 2 is possibly deleted). Then the algorithm produces an infinite sequence of points in \mathcal{L}_0 , such that there exist limit points and every limit point \bar{x} of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.*

Proof It is easily seen that all the assumptions of Proposition 19.3 are satisfied and hence the assertion follows from that proposition. \square

Remark 19.6 It is known that, in the general case, coordinate methods employing exact linesearches may not converge, but inexact searches like those considered here can guarantee, as we have seen, convergence towards stationary points under usual assumptions. The coordinate method can obviously be viewed as a decomposition method of the Gauss-Seidel type and hence additional convergence results can be found in Chap. 26. \square

Remark 19.7 Algorithm 19.5.2 is just a conceptual scheme. In particular, we could associate a different parameter ρ_k to each coordinate direction. We can also replace the bidirectional searches with positive searches with $\alpha \geq 0$ by performing $n + 1$ searches at Step 1 along a positive basis by means of Algorithm 19.5.1. \square

Globalization techniques based on nonmonotone derivative-free linesearch algorithm can be found in [131].

19.6 Approximation of Derivatives and Implicit Filtering

In order to improve the efficiency of direct search methods, it is often convenient to employ suitable approximations of the derivatives. As already remarked, a major difficulty can be the presence of noise in the objective function, so that we must avoid to take very small steps in finite differencing. At the same time, when a function evaluation is very costly and time-consuming, we would like to make use of past information for approximating derivatives at the current point. In this section we will give a short description of some the techniques that attempt to overcome, as much as possible, these difficulties.

19.6.1 Simplex Gradient

Suppose we have computed and stored $r + 1$ points $y^i \in R^n$ and the corresponding function values $f(y^i)$, for $i = 0, 1, \dots, r$. Let us define the $n \times r$ matrix V with columns v^i defined by $v^i = y^i - y^r$, $i = 0, 1, \dots, r - 1$, and the r -vector $\delta(f)$ with components

$$\delta_i(f) = f(y^i) - f(y^r), \quad i = 0, 1, \dots, r - 1.$$

Example 19.1 Consider the particular case of $y^i = y^n + \delta_i e_i$, for $i = 0, \dots, n - 1$. Then we have

$$V = \begin{pmatrix} \delta_0 & 0 & \dots & 0 \\ 0 & \delta_1 & 0 & \dots & 0 \\ \dots & & & & \\ 0 & 0 & 0 & \dots & \delta_{n-1} \end{pmatrix}$$

and

$$V^{-1}\delta(f) = \begin{pmatrix} \frac{f(y^0) - f(y^n)}{\delta_0} \\ \frac{f(y^1) - f(y^n)}{\delta_1} \\ \vdots \\ \frac{f(y^{n-1}) - f(y^n)}{\delta_{n-1}} \end{pmatrix}.$$

This latter vector can be viewed as a finite-difference approximation of the gradient. \square

If we assume that $r = n$, we know that the convex hull of the set of $n + 1$ points y^i is defined a *simplex* in R^n and, if the matrix V is nonsingular, the simplex is termed a *nonsingular simplex* and each point y^i is a *vertex* of the simplex. Then we can introduce the definition of *simplex gradient*.

Definition 19.1 Let $S \subset R^n$ be a nonsingular simplex with vertices $y^i, i = 0, 1, \dots, n$. The simplex gradient $Df(y^n)$ of f at y^n is defined by

$$Df(y^n) = V^{-T}\delta(f). \quad (19.28)$$

\square

The following proposition, which establishes a relation between the gradient and the simplex gradient, is a slightly modified version of that given in [153].

Proposition 19.7 Suppose that $S \subseteq R^n$ is a nonsingular simplex and assume that ∇f is Lipschitz continuous on an open neighborhood of S with Lipschitz constant L . Then there is a constant K depending only on L , such that

$$\|\nabla f(y^n) - Df(y^n)\| \leq K\kappa(V)\sigma_+(S),$$

where $\kappa(V)$ is the condition number of V and $\sigma_+(S) = \max_{0 \leq i \leq n-1} \|y^i - y^n\|$.

Proof By the Theorem of the Mean, letting $v^i = y^i - y^n$, for $i = 0, 1, \dots, n - 1$, we have:

$$f(y^i) = f(y^n) + \nabla f(z^i)^T v^i + \nabla f(y^n)^T v^i - \nabla f(y^n)^T v^i,$$

where $z^i = y^n + \theta^i v^i$ for some $\theta^i \in (0, 1)$. Then we can write:

$$f(y^n) - f(y^i) + \nabla f(y^n)^T v^i = (\nabla f(y^n) - \nabla f(z^i))^T v^i.$$

Letting $w_i = (\nabla f(y^n) - \nabla f(z^i))^T v^i$, for $i = 0, 1, \dots, n-1$ and $w = (w_0 \dots w_{n-1})^T$, we obtain

$$-\delta(f) + V^T \nabla f(y^n) = w.$$

Premultiplying both members by V^{-T} we have

$$-V^{-T} \delta(f) + \nabla f(y^n) = V^{-T} w,$$

so that, recalling the definition of the simplex gradient $Df(y^n)$, we can write

$$\nabla f(y^n) - Df(y^n) = V^{-T} w.$$

This implies

$$\|\nabla f(y^n) - Df(y^n)\| \leq \|V^{-1}\| \|w\| = \|V^{-1}\| \left(\sum_{i=0}^{n-1} |w_i|^2 \right)^{1/2}. \quad (19.29)$$

Now, we have that

$$|w_i| \leq \|\nabla f(y^n) - \nabla f(z^i)\| \|v^i\| \leq L \|v^i\|^2 \leq L \sigma_+^2(S).$$

Then by (19.29), we can write

$$\|\nabla f(y^n) - Df(y^n)\| \leq \|V^{-1}\| \|w\| \leq n^{1/2} L \|V^{-1}\| \sigma_+^2(S)$$

and, moreover, taking into account the fact that, by $\|\cdot\|$ we intend the Euclidean norm, we have

$$\sigma_+(S) = \max_{0 \leq i \leq n-1} \|v^i\| \leq \max_{0 \leq i \leq n-1} \|v^i\|_1 = \|V\|_1 \leq n^{1/2} \|V\|,$$

and hence, as $\kappa(V) = \|V^{-1}\| \|V\|$, the assertion is proved, by taking $K = nL$. \square

In practice, it could be convenient to define the gradient approximation even if $r \neq n$ or V is singular, by computing $g(y^r)$ as a solution of the least squares problem

$$\min_g \|V^T g - \delta(f)\|^2.$$

The gradient approximation obtained from available past information can be used for improving direct search techniques from different points of view. Here we will confine ourselves to give some information on *implicit filtering* and on linesearch-based combinations with coordinate methods.

19.6.2 Implicit Filtering

Implicit filtering algorithms have been designed to solve optimization problems, where the objective function $f : R^n \rightarrow R$, obtained through some “black box” process, is affected by the presence of noise, that is we have

$$f(x) = \hat{f}(x) + \phi(x),$$

where \hat{f} is a continuously differentiable function and ϕ is a “small” random noise, which may introduce many small oscillations that create several spurious local minimizers.

We suppose that the computed function values are not filtered directly and we attempt to extract, as much as possible, the information on the (unknown) gradient of \hat{f} .

In the simplest form, the method can be viewed as a modified version of the gradient method, where a backtracking Armijo-type linesearch is performed along a finite difference approximation of the negative gradient with a finite difference step which is not too small. The main feature of this approach is that if the line search “fails”, i.e., after a maximum number of trials a standard condition of sufficient reduction is not satisfied, the step ϵ is reduced and the procedure is repeated. The difference increment ϵ can be reduced as the iteration progresses until some tolerance on ϵ is reached.

The gradient approximation used in this approach can be obtained by employing the simplex gradient defined before, by assuming that the diameter of the simplex is reduced in case of failure.

The method may be particularly convenient when the level of noise decreases in a neighborhood of the solution. Therefore, the method can be advantageously adopted when the level of noise can be managed. This may happen, for instance, when the evaluation of the objective function f is obtained as the result of a numerical procedure (similar to that used for numerically solving, for instance, a differential equation) where the degree of precision, which can be arbitrarily defined, determines the level of noise (lower levels of noise are obtained by higher accuracies).

19.6.3 Combination with Coordinate Search

An alternative approach to that described above can be that of combining a cycle of coordinate searches with an acceleration step based on a search along an approximated gradient. In particular, we can consider a conceptual scheme where each major iteration is performed in two steps.

Step (a). At each major step k , starting from $y^0 = x_k$, a cycle of bidirectional derivative free Armijo-type line searches is carried out along the coordinate axes or, more generally, along a set of directions d^i , for $i = 1, \dots, q$ with a given tolerance. During this phase, we further store $r + 1$ points $y^i \in R^n$ and the corresponding function values $f(y^i)$, for $i = 0, 1, \dots, r$.

Step (b). We first compute a simplex gradient g (in an extended sense) as the solution $g(y^r)$ of the linear least squares problem of minimizing $\|V^T g - \delta(f)\|^2$, where V and $\delta(f)$ are defined as in Sect. 19.6.1. Then we perform a derivative-free line search along $d = -g(y^r)$ and we set $x_{k+1} = y^r + \alpha d$, where possibly $\alpha = 0$. We set $k = k + 1$ and repeat Step (a) with a reduced tolerance if $\alpha = 0$.

Since a coordinate search is performed within a finite number of iterations, starting from each x_k , it can be easily verified that the algorithm converges, provided that the assumptions of Proposition 19.3 or Proposition 19.5 are satisfied.

19.7 Model-Based Methods

The idea underlying the methods here briefly described is to build at each iteration an analytical model (typically a quadratic model) of the objective function interpolating the function values attained in correspondence to points suitably generated. Let x_k be the current point and let $Y = [y_1, \dots, y_p]$ a set of points in R^n where the objective function has been evaluated.

Consider a quadratic model

$$m_k(x_k + s) = f(x_k) + b^T s + \frac{1}{2} s^T Q s \quad (19.30)$$

where $b \in R^n$ and Q is a symmetric $n \times n$ matrix. The number of unknown coefficients of the model, represented by the elements of vector b and of the matrix Q , is $n + \frac{n(n+1)}{2}$. These coefficient can be determined by imposing the interpolation conditions

$$m_k(y_i) = f(y_i) \quad i = 1, \dots, p. \quad (19.31)$$

The above conditions uniquely define the model m_k only if $p = n + \frac{n(n+1)}{2}$. In this case condition (19.31) define a square linear system whose unknowns are the coefficients of the model.

In general the model m_k will be nonconvex. Therefore, it is reasonable to exploit it with a trust region strategy. In particular, a (possibly approximated) solution s_k of the subproblem

$$\begin{aligned} \min m_k(x_k + s) &= f(x_k) + b^T s + \frac{1}{2} s^T Q s \\ \|s\| &\leq \Delta_k. \end{aligned} \tag{19.32}$$

is determined, where $\Delta > 0$ is the trust region radius. The trial point $x_k + s_k$ is accepted as the new point x_{k+1} provided that s_k determines a “sufficient reduction” of the quadratic model $m_k(s)$ and such a reduction implies a sufficient reduction of the objective function computed as $f(x_k + s_k) - f(x_k)$.

Whenever a “sufficient reduction” has not been attained, possible causes may be the following:

- (i) the geometrical distribution of the interpolation set Y is “not optimal”;
- (ii) the amplitude of the radius Δ is “too high”.

Issue (i) is checked evaluating the condition number of the matrix of the linear system defined by the interpolation conditions: if this number is “too big” then an element of Y is replaced by a new point in order to obtain a better conditioned matrix. There exist several techniques to update the interpolation set Y (which requires to choose the point to be eliminated and the point to be inserted) and to update the model m_k . In particular, in order to reduce the computational burden, the model m_k is updated and not recomputed.

When issue (i) is not occurred, the radius Δ_k is simply reduced as in the trust region strategy already seen.

Finally, we observe that the employment of the quadratic model could be too computationally heavy. In particular, the only initialization of the algorithm requires $n + \frac{n(n+1)}{2}$ function evaluations, and this could be prohibitive even if the dimension n of the problem is not too high. In order to overcome this issue, it could be suitable to use a linear model (setting Q equal to the null matrix) at the early iterations and to switch to a quadratic model when the function has been evaluated at a sufficient number of points.

19.8 Exercises

19.1 Given the quadratic function

$$f(x) = \frac{1}{2}x^T Qx + c^T x + d,$$

where Q is a symmetric $n \times n$ matrix, show that, for every $t > 0$ we have

$$\frac{\partial f(x)}{\partial x_j} = \frac{f(x + te_j) - f(x - te_j)}{2t}.$$

19.2 Define a computer code of the coordinate method with simple decrease and perform some computational experiments.

19.3 Define a computer code of the coordinate method with derivative-free Armijo-Goldstein linesearch and perform some computational experiments.

19.4 Define a computer code of the Hooke-Jeeves method with derivative-free Armijo-Goldstein linesearch and perform some computational experiments.

19.9 Notes and References

Suggested general references on (monotone) derivative-free methods are the paper [157] and the books [50], [206]. The Hooke-Jeeves method has been introduced in [144]. A class of methods (called *pattern search* methods), that includes the coordinate method with simple decrease and the Hooke-Jeeves method is defined in [251], where more general global convergence results than those presented in the chapter can be found. The results on the coordinate method with simple decrease reported in the chapter are based essentially on the above paper. Rosenbrock method, cited in the chapter, is described in [230] and [12]. Applications to the globalization of Rosenbrock method can be found in [131]. The Nelder-Mead method has been proposed in [190]. As shown in [184], the method can fail to converge even in problems with two variables. Different modified versions of the Nelder-Mead method with global convergence properties have been proposed in the literature. (see, for instance [154] and [252]). The coordinate methods based on sufficient decrease have been defined in [123], [175], and [131]. Suggested references on model based methods can be the books [51], [196], and the paper [219]. The use of derivative-free conjugate gradient method was studied in [211]. As regards implicit filtering algorithms, the main references are [48] and [154]. The paper [53] is a useful reference to analyze the possibility of improving direct search techniques using the gradient approximation obtained from available past information.

Chapter 20

Methods for Problems with Convex Feasible Set



In this chapter we consider constrained optimization problems, where the feasible set is a convex set, and we describe methods that preserve feasibility, starting from a given feasible point. In particular, we consider optimization problems whose objective function is a general nonlinear continuously differentiable function. Specialized feasible methods for Linear Programming and Quadratic Programming are out of the scope of this book and the interested reader should refer to the literature cited in Chap. 1.

Preliminarily we recall the optimality conditions given in Chap. 4 and we describe an Armijo-type line search employed by the algorithms defined here. Then we present two gradient-based methods: the *Frank-Wolfe method*, and the simplest version of the *gradient projection method*. These methods can be viewed as extensions of the unconstrained gradient method to the convex constrained case. With reference to problems with box constraints, we define a derivative-free algorithm based on the coordinate directions.

Finally, we consider a sparse optimization problem concerning the minimization of the zero-norm of a vector over a polyhedral set, we present a concave optimization-based approach and we state finite convergence results for the Frank-Wolfe method applied to the specific concave programming problem.

20.1 Problems with Convex Feasible Set

Consider the problem

$$\min f(x), \quad x \in S,$$

where $f : R^n \rightarrow R$ is a continuously differentiable function and $S \subset R^n$ is a convex set.

We recall from Chap. 4 the following first order optimality condition.

Proposition 20.1 (Optimality Condition) *Let $x^* \in S$ be a local minimum point of the problem*

$$\min f(x), \quad x \in S,$$

where $S \subset R^n$ is a convex set. Suppose that f is continuously differentiable in a neighborhood of x^ . Then we have*

$$\nabla f(x^*)^T(x - x^*) \geq 0, \quad \text{for all } x \in S. \quad (20.1)$$

□

We say that $x^* \in S$ is a *critical point* if the necessary condition of Proposition 20.1 holds at x^* .

It can be easily shown that condition (20.1) is a *necessary and sufficient condition of global minimum* if we assume that f is convex. We recall also the definition of *projection of a point x on a convex set* and the characterization of the projection.

Definition 20.1 (Projection of a Point on a Convex Set) Let $S \subseteq R^n$ be a closed, nonempty convex set and let $x \in R^n$ be a given point. The projection of x on S is the solution $p(x)$ of the following problem

$$\min \{\|x - y\|, \quad y \in S\},$$

that is, $p(x) \in S$ is the point such that

$$\|x - p(x)\| \leq \|x - y\|, \quad \text{for all } y \in S.$$

□

Proposition 20.2 (Characterization of the Projection) *Let $S \subseteq R^n$ be a convex, nonempty set, let $x \in R^n$ be a given point and let $\|\cdot\|$ be the Euclidean norm. Then*

- (i) *a point $y^* \in S$ is the projection of x on S , i.e., $y^* = p(x)$ if and only if*

$$(x - y^*)^T (y - y^*) \leq 0, \quad \text{for all } y \in S;$$

- (ii) *the projection mapping is continuous and non-expansive, that is*

$$\|p(x) - p(z)\| \leq \|x - z\| \quad \text{for all } x, z \in R^n.$$

□

The optimality condition of Proposition 20.1 can be restated in terms of projection.

Proposition 20.3 *Consider the problem $\min f(x)$, $x \in S$, where $S \subseteq R^n$ is a convex set. Let $x^* \in S$ and suppose that f is continuously differentiable in a neighborhood of x^* . Then the point x^* is a critical point if and only if*

$$x^* = p[x^* - s \nabla f(x^*)], \quad (20.2)$$

for some $s > 0$.

□

Therefore, when we consider a critical point we can equivalently refer either to condition (20.1) or to condition (20.2).

20.2 Line Search Along a Feasible Direction

We will consider algorithms based on line searches along feasible directions which can be viewed as simple extensions of the line searches presented in the case of unconstrained optimization. We assume that the following conditions hold.

- (a) S is a convex set;
- (b) for every k the point $x_k + d_k$ belongs to S (and hence, thanks to the convexity of S , the direction d_k is a feasible direction for S at x_k);
- (c) for every k the direction d_k is a descent direction such that $\nabla f(x_k)^T d_k < 0$.

Under the above assumptions, we can define a line search algorithm that produces feasible points. For instance, we can define the following Armijo-type line search using $\alpha = 1$ as initial step-size.

Armijo Line Search

Data: $\gamma \in (0, 1)$, $\delta \in (0, 1)$.

Set $\alpha = 1$ e $j = 0$.

While $f(x_k + \alpha d_k) > f(x_k) + \gamma \alpha \nabla f(x_k)^T d_k$

set $\alpha = \delta \alpha$ and $j = j + 1$.

End While

Set $\alpha_k = \alpha$ and terminate.

□

Reasoning as in the unconstrained case we can prove that, under assumptions (a), (b), (c), if $x_k \in S$, the Armijo line search determines in a finite number of iterations a value $\alpha_k \in (0, 1]$ such that $x_{k+1} \in S$ and

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k < f(x_k). \quad (20.3)$$

We can state the following convergence result similar to that established in the unconstrained case.

Proposition 20.4 (Convergence of Armijo Line Search) *Let $f : R^n \rightarrow R$ be a continuously differentiable function on an open set containing S . Assume that S is a convex and compact set and suppose that $\{x_k\}$ is an infinite sequence of points belonging to S such that $x_0 \in S$ and for every k we have $x_k + d_k \in S$ and $\nabla f(x_k)^T d_k < 0$. Then the Armijo line search determines in a finite number of iterations a step-size $\alpha_k \in (0, 1]$ such that the sequence $x_{k+1} = x_k + \alpha_k d_k$ satisfies the following conditions*

- (c₁) $x_{k+1} \in S$;
- (c₂) $f(x_{k+1}) < f(x_k)$;
- (c₃) $\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k = 0$.

Proof As already observed, it can be easily verified that the Armijo line search determines in a finite number of iterations a step-size $\alpha_k \in (0, 1]$ such that (c₁) and (c₂) hold. We will prove that even (c₃) is satisfied.

First we observe that, as $x_k \in S$ and $x_k + d_k \in S$ for every k , from the compactness of S we get that the sequence $\{\|d_k\|\}$ is limited. Therefore, we have that there exists $M > 0$ such that $\|d_k\| \leq M$ for every k .

Since α_k is such that condition (20.3) holds, we can write

$$f(x_k) - f(x_{k+1}) \geq \gamma \alpha_k |\nabla f(x_k)^T d_k|. \quad (20.4)$$

The sequence $\{f(x_k)\}$ is a monotonic decreasing sequence, furthermore it is bounded below (since f is continuous over the compact set S), so that the limit $\{f(x_k)\}$ for $k \rightarrow \infty$ exists and we have

$$\lim_{k \rightarrow \infty} \alpha_k |\nabla f(x_k)^T d_k| = 0. \quad (20.5)$$

Now, by contradiction, let us assume that (c_3) is not true. The sequence $\{\nabla f(x_k)^T d_k\}$ is limited, and hence there exists a subsequence (relabelled $\{x_k\}$), such that

$$\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k = -\eta < 0, \quad (20.6)$$

where η is a positive number. From (20.5) it follows

$$\lim_{k \rightarrow \infty} \alpha_k = 0. \quad (20.7)$$

Since $x_k \in S$ (which is a compact set) and the sequence $\{d_k\}$ is limited, there exist subsequences (relabelled by $\{x_k\}$ and $\{d_k\}$), such that

$$\lim_{k \rightarrow \infty} x_k = \hat{x} \in S, \quad \lim_{k \rightarrow \infty} d_k = \hat{d}. \quad (20.8)$$

From (20.6) and (20.8), taking into account the continuity of ∇f , we obtain

$$\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k = \nabla f(\hat{x})^T \hat{d} = -\eta < 0. \quad (20.9)$$

Using (20.7). for k sufficiently large, say $k \geq \hat{k}$, we have $\alpha_k < 1$ and hence, for $k \geq \hat{k}$, we can write

$$f(x_k + \frac{\alpha_k}{\delta} d_k) - f(x_k) > \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k. \quad (20.10)$$

Using the mean value theorem we have

$$f(x_k + \frac{\alpha_k}{\delta} d_k) = f(x_k) + \frac{\alpha_k}{\delta} \nabla f(z_k)^T d_k, \quad (20.11)$$

with

$$z_k = x_k + \theta_k \frac{\alpha_k}{\delta} d_k \quad \text{where } \theta_k \in (0, 1).$$

Replacing (20.11) by (20.10), for $k \geq \hat{k}$, we obtain

$$\nabla f(z_k)^T d_k > \gamma \nabla f(x_k)^T d_k. \quad (20.12)$$

Since $\|d_k\| \leq M$, from (20.7) it follows $\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0$ and hence

$$\lim_{k \rightarrow \infty} z_k = \lim_{k \rightarrow \infty} \left(x_k + \theta_k \frac{\alpha_k}{\delta} d_k \right) = \hat{x}.$$

As a consequence, taking the limits for $k \rightarrow \infty$, from (20.12) we get

$$\nabla f(\hat{x})^T \hat{d} \geq \gamma \nabla f(\hat{x})^T \hat{d}.$$

The above condition, taking into account (20.9), implies $\eta \leq \gamma \eta$ and this contradicts the assumption $\gamma < 1$. Then we can conclude that (20.9) leads to a contradiction, so that (c₃) must hold. \square

As already observed in the unconstrained case, in the proof of the preceding proposition it is not necessary to impose that $x_{k+1} = x_k + \alpha_k^A d_k$, being α_k^A the step-size computed by the Armijo line search. It is sufficient to get x_{k+1} such that

$$f(x_{k+1}) \leq f(x_k + \alpha_k^A d_k),$$

provided that x_{k+1} belongs to S . This implies that suitable values for α are those such that $\alpha \in (0, 1]$ and $f(x_k + \alpha d_k)$ is lower or equal than $f(x_k + \alpha_k^A d_k)$. Therefore, Proposition 20.4 shows convergence properties of a line search where the step-size α_k is computed in a such a way that

$$f(x_k + \alpha_k d_k) = \min_{\alpha \in [0, 1]} f(x_k + \alpha d_k).$$

From the preceding results it follows that, in order to guarantee the convergence towards critical points by using feasible directions, it is sufficient to show that the direction d_k is a feasible and descent direction and that the limit $\nabla f(x_k)^T d_k \rightarrow 0$ implies the convergence towards critical points.

Two choices of d_k that allow us to satisfy the above properties are presented in the next sections.

20.3 The Frank-Wolfe Method (Conditional Gradient Method)

Let $S \subset R^n$ be a nonempty compact and convex set and let f be a continuously differentiable function. We consider the problem

$$\min f(x), \quad x \in S,$$

with the aim of defining an algorithm with convergence properties towards critical points of f on S .

Given a point $x_k \in S$, we can try to define a feasible and descent direction at x_k by solving the convex programming problem with linear objective function

$$\begin{aligned} \min & \nabla f(x_k)^T (x - x_k) \\ & x \in S \end{aligned} \tag{20.13}$$

Since S is compact the above problem admits solution $\hat{x}_k \in S$. If the optimal value is zero, that is $\nabla f(x_k)^T (\hat{x}_k - x_k) = 0$, we have

$$0 = \nabla f(x_k)^T (\hat{x}_k - x_k) \leq \nabla f(x_k)^T (x - x_k), \quad \text{for all } x \in S,$$

and hence x_k is, by definition, a critical point.

If $\nabla f(x_k)^T (\hat{x}_k - x_k) < 0$, then we can consider the direction $d_k = \hat{x}_k - x_k$, which is a feasible and descent direction at x_k and we can define the iteration

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

where $\alpha_k \in (0, 1]$ can be determined by an Armijo-type line search. Thanks to the convexity of S , if $\alpha_k \in [0, 1]$ then the point x_{k+1} belongs to S .

The algorithm is known as the *Method of Frank-Wolfe* or *Conditional gradient method*. The formal conceptual description of the method is reported below.

Method of Frank-Wolfe

1. Choose the initial point $x_0 \in S$.
- For k=0,1,...
2. Compute a solution \hat{x}_k of problem (20.13) and set $d_k = \hat{x}_k - x_k$; if $\nabla f(x_k)^T d_k = 0$ terminate.
3. Compute a step-size $\alpha_k > 0$ along d_k by the Armijo line search.
4. Set $x_{k+1} = x_k + \alpha_k d_k$.
- End For

Fig. 20.1 One iteration of the method of Frank-Wolfe

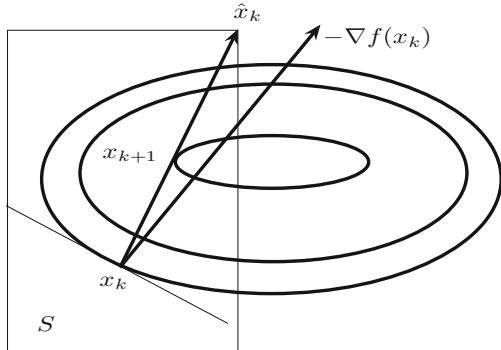


Figure 20.1 shows one iteration of the method in a problem in two variables, assuming the at each iteration α_k is the minimizer of $f(x_k + \alpha d_k)$ for $\alpha \in [0, 1]$. We can state a convergence result of the method of Frank-Wolfe.

Proposition 20.5 (Convergence of the Method of Frank-Wolfe) *Let $f : R^n \rightarrow R$ be continuously differentiable over an open set containing the non empty compact convex set S . Let $\{x_k\}$ be the sequence produced by the method of Frank-Wolfe. Then, either there exists an index $v \geq 0$ such that x_v is a critical point, or an infinite sequence is generated and each accumulation point of $\{x_k\}$ is a critical point.*

Proof Recalling the convergence results of the Armijo line search, it can be easily verified that, by assuming that the algorithm does not terminate in a finite number of iterations at a critical point, we have $f(x_{k+1}) < f(x_k)$ for every k and we can write

$$\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k = 0.$$

The compactness of S implies that there exists an accumulation point $\bar{x} \in S$; moreover, the direction d_k is bounded being

$$\|d_k\| = \|\hat{x}_k - x_k\| \leq \|\hat{x}_k\| + \|x_k\|,$$

with $x_k, \hat{x}_k \in S$. Then, there exists a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K, k \rightarrow \infty} x_k = \bar{x}, \quad \lim_{k \in K, k \rightarrow \infty} d_k = \bar{d}.$$

It follows that

$$\nabla f(\bar{x})^T \bar{d} = 0.$$

From the definition of d_k we get

$$\nabla f(x_k)^T d_k \leq \nabla f(x_k)^T (x - x_k) \quad \text{for all } x \in S,$$

so that, taking the limits for any given $x \in S$, we obtain

$$0 = \nabla f(\bar{x})^T \bar{d} \leq \nabla f(\bar{x})^T (x - \bar{x}),$$

that is

$$\nabla f(\bar{x})^T (x - \bar{x}) \geq 0, \quad \text{for all } x \in S,$$

and this proves that \bar{x} is a critical point of f , and the thesis is proved \square

Suppose that the feasible set is defined by linear constraints, that is

$$S = \{x \in R^n : Ax \geq b\};$$

in this case the subproblem for computing the search direction is a *linear programming problem*, i.e.,

$$\min \nabla f(x_k)^T x$$

$$Ax \geq b,$$

whose solution \hat{x}_k defines the search direction $d_k = \hat{x}_k - x_k$.

In the general case the method of Frank-Wolfe may be inefficient and may show a sublinear rate of convergence. However, the method can be advantageously applied in large-scale problems concerning network optimization, whenever an high degree of precision is not required and the constraints have a particular structure that leads to the possibility of efficiently computing the search direction d_k .

Finally, we observe that the method requires the knowledge of a feasible initial point x_0 . Then, in the case that a feasible point is not available, a feasibility problem must be solved.

20.4 Gradient Projection Method

A different way to extend the unconstrained steepest descent method to the case of constrained optimization is represented by the *gradient projection method*.

The method is defined by an iteration of the form

$$x_{k+1} = x_k + \alpha_k (p[x_k - s_k \nabla f(x_k)] - x_k),$$

where $\alpha_k \in (0, 1]$, $s_k > 0$ and $p[x_k - s_k \nabla f(x_k)] \in S$ is the projection of the (usually unfeasible) point generated along the negative gradient

$$x_k - s_k \nabla f(x_k).$$

It can be easily verified that the direction

$$d_k = p[x_k - s_k \nabla f(x_k)] - x_k$$

is a feasible direction at x_k . We observe that in the unconstrained case, that is, when $S = R^n$, we have $p[x_k - s_k \nabla f(x_k)] = x_k - s_k \nabla f(x_k)$ and, as a consequence, we obtain the negative gradient direction $d_k = -s_k \nabla f(x_k)$.

Now we prove that d_k is a descent direction, provided that $d_k \neq 0$.

Proposition 20.6 *Let $f : R^n \rightarrow R$ be continuously differentiable over an open set containing the compact convex set S . Let*

$$d_k = p[x_k - s_k \nabla f(x_k)] - x_k,$$

where $s_k > 0$, and $p[x_k - s_k \nabla f(x_k)] \in S$ is the projection of the point

$$x_k - s_k \nabla f(x_k)$$

on the set S . Then, if $d_k \neq 0$ we have

$$\nabla f(x_k)^T d_k < 0. \quad (20.14)$$

Proof Set

$$\hat{x}_k = p[x_k - s_k \nabla f(x_k)],$$

so that we have $d_k = \hat{x}_k - x_k$. Recalling the properties of the projection mapping, we must have

$$(x_k - s \nabla f(x_k) - \hat{x}_k)^T (x - \hat{x}_k) \leq 0, \quad \text{for all } x \in S,$$

from which we obtain, setting $x = x_k$,

$$(x_k - s_k \nabla f(x_k) - \hat{x}_k)^T (x_k - \hat{x}_k) \leq 0.$$

Therefore we have

$$\nabla f(x_k)^T d_k = \nabla f(x_k)^T (\hat{x}_k - x_k) \leq -\frac{1}{s_k} \|x_k - \hat{x}_k\|^2, \quad (20.15)$$

and this proves that d_k is a descent direction at x_k provided that $\|x_k - \hat{x}_k\| \neq 0$. \square

The gradient projection method can be realized both fixing s_k to a constant value and performing an Armijo-type line search to compute α_k , and fixing α_k to a constant value in $(0, 1]$ and performing a search on s_k (in the latter case, by varying s_k , a curvilinear path is defined on S).

Here we consider only the case $s_k = s > 0$, assuming that S is a convex, compact and nonempty set. The formal scheme of the algorithm is reported below.

Gradient Projection Method

1. Choose the initial point $x_0 \in S$, and a scalar $s > 0$.

For k=0,1,...

2. Compute

$$\hat{x}_k = p[x_k - s \nabla f(x_k)],$$

if $\hat{x}_k = x_k$ terminate; otherwise, set $d_k = \hat{x}_k - x_k$.

3. Compute the step-size $\alpha_k > 0$ along d_k by the Armijo line search.
4. Set $x_{k+1} = x_k + \alpha_k d_k$.

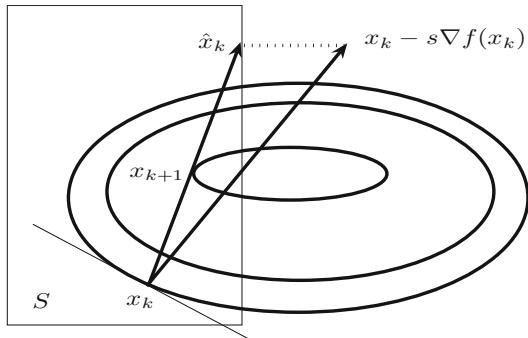
End For

One iteration of the method, with exact line search, is shown in Fig. 20.2.

We can state the following convergence result of the gradient projection method.

Proposition 20.7 (Convergence of the Gradient Projection Method) *Let $f : R^n \rightarrow R$ be continuously differentiable over an open set containing the non empty compact convex set S . Let $\{x_k\}$ be the sequence generated by the gradient projection method. Then, either there exists an index $v \geq 0$ such that x_v is a critical point, or an infinite sequence is generated and each accumulation point of $\{x_k\}$ is a critical point.*

Fig. 20.2 One iteration of the gradient projection method



Proof If the algorithm terminates in a finite number of iterations, say at iteration v , then we have

$$x_v = p[x_v - s \nabla f(x_v)],$$

so that, using Proposition 20.3, we can conclude that x_v is a critical point.

Let us consider an infinite sequence of points $\{x_k\}$ generated by the algorithm starting from the initial point $x_0 \in S$. Set

$$\hat{x}_k = p[x_k - s \nabla f(x_k)]$$

and $d_k = \hat{x}_k - x_k$. The direction d_k is a feasible direction, furthermore, from Proposition 20.6 we get that d_k is even a descent direction at x_k provided that $\|x_k - \hat{x}_k\| \neq 0$.

Since the step-size α_k along d_k is computed by the Armijo line search, from known results we have that $f(x_{k+1}) < f(x_k)$ for every k and

$$\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k = 0. \quad (20.16)$$

The compactness of S implies the existence of an accumulation point $\bar{x} \in S$ of $\{x_k\}$. Therefore, we can define a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K, k \rightarrow \infty} x_k = \bar{x}.$$

From (20.15) and (20.16) it follows

$$\lim_{k \in K, k \rightarrow \infty} \|p[x_k - s \nabla f(x_k)] - x_k\| = 0.$$

Then, recalling the continuity of the projection mapping, taking the limits we obtain

$$p[\bar{x} - s \nabla f(\bar{x})] = \bar{x},$$

and this proves that \bar{x} is a critical point. \square

The gradient projection method needs to perform the projection operation. This may cause difficulties whenever the projection of a point on the feasible set can not be computed efficiently. We observe that even in the case of linear constraints the projection operation can be costly, since it requires to solve a quadratic programming problem. The gradient projection method has convergence rate better than the method of Frank-Wolfe. Then, the former is suitable whenever the projection operation can be performed efficiently.

In the case of simple constraints, there are versions of the gradient projection method where s_k is determined with the Barzilai-Borwein method (see Chap. 25), and the step-size α_k is computed by a nonmonotone Armijo-type line search. These nonmonotone methods seem to be promising from a computational point of view.

20.5 A Derivative-Free Method for Box Constrained Problems

Let us consider a problem of the form

$$\begin{aligned} & \min f(x) \\ & Ax \leq b \end{aligned} \tag{20.17}$$

where $f : R^n \rightarrow R$ is a continuously differentiable function, A is a $m \times n$ matrix, $b \in R^m$. We assume that the gradient of the objective function is not available.

As already seen, a derivative-free method must necessarily employ at each iteration a set of search directions. Indeed, without gradient information, it is not possible to ensure that a single feasible direction is a descent direction. The minimal requirement of a set of directions is that it contains at least a feasible and descent direction. We will see that, in the specific case of box constraints, this requirement is satisfied by the set of the coordinate directions. First we state results in the general case of linear inequality constraints.

We denote by S the feasible set of (20.17), i.e., $S = \{x \in R^n : Ax \leq b\}$. Given a point $\bar{x} \in S$, we indicate by $I_0(\bar{x})$ the set of indices of active constraints, i.e.,

$$I_0(\bar{x}) = \{i \in \{1, \dots, m\} : a_i^T \bar{x} = b_i\}$$

As shown by Proposition 4.4, the set of feasible directions at \bar{x} is

$$D(\bar{x}) = \{d \in R^n : a_i^T d \leq 0 \quad \text{for all } i \in I_0(\bar{x})\} \tag{20.18}$$

In the particular case of box constraints, i.e., $S = \{x \in R^n : l \leq x \leq u\}$, we have

$$D(\bar{x}) = \{d \in R^n : d_i \geq 0 \quad \text{for all } i \in L(\bar{x}), d_i \leq 0 \quad \text{for all } i \in U(\bar{x})\},$$

where

$$L(\bar{x}) = \{i \in \{1, \dots, n\} : \bar{x}_i = l_i\} \quad U(\bar{x}) = \{i \in \{1, \dots, n\} : \bar{x}_i = u_i\}.$$

It is easy to see that a feasible point \bar{x} is a critical point, i.e.,

$$\nabla f(\bar{x})^T(x - \bar{x}) \geq 0 \quad \text{for all } x \in S,$$

if and only if

$$\nabla f(\bar{x})^T d \geq 0 \quad \text{for all } d \in D(\bar{x}).$$

We will consider subsequences $\{x_k\}$ of feasible points convergent to some limit point \bar{x} . It is important to analyze the relationship between $D(\bar{x})$ and $D(x_k)$. To this aim we first prove the following result.

Proposition 20.8 *Let $\{x_k\}$ be a sequence of feasible points such that $x_k \rightarrow \bar{x}$ for $k \rightarrow \infty$. Then for k sufficiently large we have*

$$I_0(x_k) \subseteq I_0(\bar{x}).$$

Proof By contradiction, let us assume that for every integer $\tilde{k} > 0$ we can find an integer $k > \tilde{k}$ such that $i_k \in I_0(x_k)$ and $i_k \notin I_0(\bar{x})$. Since $i_k \in \{1, \dots, m\}$ there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that, for some i we have $i_k = i$ for all $k \in K$. Then we can write

$$a_i^T x_k - b_i = 0 \quad \text{and} \quad a_i^T \bar{x} - b_i < 0.$$

Taking the limits for $k \in K$ and $k \rightarrow \infty$ we obtain

$$a_i^T \bar{x} - b_i = 0 \quad \text{and} \quad a_i^T \bar{x} - b_i < 0,$$

i.e., a contradiction. □

From the preceding proposition we immediately get the following result.

Proposition 20.9 *Let $\{x_k\}$ be a sequence of feasible points such that $x_k \rightarrow \bar{x}$ for $k \rightarrow \infty$. Then for k sufficiently large we have*

$$D(\bar{x}) \subseteq D(x_k).$$

Proof Consider any $d \in D(\bar{x})$. By (20.18) we have

$$a_i^T d \leq 0 \quad \text{for all } i \in I_0(\bar{x}).$$

From Proposition 20.8 it follows for k sufficiently large

$$a_i^T d \leq 0 \quad \text{for all } i \in I_0(x_k),$$

i.e. $d \in D(x_k)$. □

Finally we state the following important property.

Proposition 20.10 Let $\{x_k\}$ be a sequence of feasible points such that $x_k \rightarrow \bar{x}$ for $k \rightarrow \infty$. Let $\bar{d} \in D(\bar{x})$. Then there exists $\hat{\beta} \in (0, +\infty]$ such that for k sufficiently large we have

$$x_k + \beta \bar{d} \in S \quad \text{for all } \beta \in [0, \hat{\beta}] \quad (20.19)$$

Proof Let

$$H = \{i \in \{1, \dots, m\} : a_i^T \bar{d} > 0\}.$$

The maximum feasible step-size $\bar{\beta}$ along \bar{d} at \bar{x} is

$$\bar{\beta} = \begin{cases} +\infty & \text{if } H = \emptyset \\ \min_{i \in H} \frac{b_i - a_i^T \bar{x}}{a_i^T \bar{d}} & \text{otherwise} \end{cases}$$

Assume first $H = \emptyset$. Then (20.19) holds with $\hat{\beta} = +\infty$.

Now assume $H \neq \emptyset$. By definition we have $\bar{\beta} > 0$ and we can write

$$a_i^T \bar{x} + \beta a_i^T \bar{d} < b_i \quad \text{for all } i \in H, \quad \text{for all } \beta \in [0, \bar{\beta}/2],$$

and hence there exists $\epsilon > 0$ such that

$$a_i^T \bar{x} + \beta a_i^T \bar{d} \leq b_i - \epsilon \quad \text{for all } i \in H, \quad \text{for all } \beta \in [0, \bar{\beta}/2]. \quad (20.20)$$

Note that

$$a_j^T \bar{x} + \beta a_j^T \bar{d} \leq b_j \quad \text{for all } j \notin H, \quad \text{for all } \beta \in [0, +\infty]. \quad (20.21)$$

As $x_k \rightarrow \bar{x}$, for k sufficiently large we have

$$|a_i^T x_k - a_i^T \bar{x}| \leq \epsilon/2. \quad (20.22)$$

Then, using (20.20) and (20.22), for k sufficiently large and for $i \in H$ we can write

$$\begin{aligned} a_i^T x_k + \beta a_i^T \bar{d} &= a_i^T x_k - a_i^T \bar{x} + a_i^T \bar{x} + \beta a_i^T \bar{d} \\ &\leq |a_i^T x_k - a_i^T \bar{x}| + a_i^T \bar{x} + \beta a_i^T \bar{d} \\ &\leq \epsilon/2 + b_i - \epsilon = b_i - \epsilon/2 \quad \text{for all } \beta \in [0, \bar{\beta}/2]. \end{aligned}$$

Finally, recalling (20.21), we obtain that (20.19) holds with $\hat{\beta} = \bar{\beta}/2$. \square

Now we consider the specific case of problems with box constraints, i.e., problems of the form

$$\begin{aligned} \min f(x) \\ l \leq x \leq u, \end{aligned} \quad (20.23)$$

where $-\infty < l_i < u_i < +\infty$ for $i = 1, \dots, n$. We indicate by $S = \{x \in R^n : l \leq x \leq u\}$ the feasible set. For any finite set $V = \{v_1, v_2, \dots, v_r\} \subset R^n$, we denote $\text{cone}\{V\} = \{v \in R^n : v = \beta_1 v_1 + \beta_2 v_2 + \dots + \beta_r v_r, \beta_1, \dots, \beta_r \geq 0\}$.

The approach here presented can easily include the possibility that some of the variables are unbounded by permitting both $l_i = -\infty$ and $u_i = +\infty$. However, for simplicity, we do not consider this possibility. In the case of box constraints we can show that the set of the coordinate directions “captures the geometry of the feasible set”. Indeed, it is possible to show that the set

$$D = \{e_1, e_2, \dots, e_n, -e_1, -e_2, \dots, -e_n\} \quad (20.24)$$

is such that the following condition

$$\text{cone}\{D \cap D(x)\} = D(x) \quad \text{for all } x \in S \quad (20.25)$$

holds. Note that the above condition implies that at any feasible point which is not a critical point the prefixed set of directions D contains at least a feasible and descent direction.

Proposition 20.11 *Let $x \in S$ and let D be the set defined by (20.24). Then*

$$\text{cone}\{D \cap D(x)\} = D(x).$$

Proof Given $x \in S$, let us consider any $d \in D(x)$. We can write

$$d = \sum_{i \in I} d_i e_i + \sum_{j \in J} |d_j|(-e_j), \quad (20.26)$$

where $I = \{i \in \{1, \dots, n\} : d_i > 0\}$, $J = \{i \in \{1, \dots, n\} : d_i < 0\}$. Since $d \in D(x)$ we have

$$x_i < u_i \quad i \in I$$

$$x_j > l_j \quad j \in J$$

so that

$$e_i \in D(x) \quad \text{for all } i \in I \quad \text{and} \quad -e_j \in D(x) \quad \text{for all } j \in J. \quad (20.27)$$

Then (20.26) and (20.27) imply the thesis. \square

We can define the following algorithm where, at iteration k , for $i = 1, \dots, 2n$:

- $\bar{\alpha}_k^i$ is the maximum feasible step-size along the direction d_i starting from x_k (if $d_i = e_i$ then $\bar{\alpha}_k^i = u_i - e_i^T x_k$; if $d_i = -e_i$ then $\bar{\alpha}_k^i = e_i^T x_k - l_i$);
- $\tilde{\alpha}_k^i > 0$ is the initial tentative step-size along d_i ;
- $\alpha_k^i \geq 0$ is the actual step-size along d_i .

If the trial step-size $\tilde{\alpha}_k^i$ is such that a condition of sufficient reduction is satisfied then an *expansion* procedure is performed to determine the step-size $\alpha_k^i \geq \tilde{\alpha}_k^i$. Otherwise, the trial step-size is reduced setting $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$, with $\theta \in (0, 1)$, and α_k^i is set equal to zero. Note that the presence of box constraints imposes restrictions on the procedure for generating the actual step-size α_k^i . In particular, the expansion procedure must be stopped whenever the new trial step-size $\delta \alpha_k^i$ is greater than the maximum feasible step-size $\bar{\alpha}_k^i$.

Coordinate Method for Box Constrained Optimization (MC-Box)

Data: set D given by (20.24); starting point $x_0 \in S$, initial step-sizes $\tilde{\alpha}_0^1, \tilde{\alpha}_0^2, \dots, \tilde{\alpha}_0^{2n} > 0$, $\theta \in (0, 1)$, $\gamma \in (0, 1)$, $\delta > 1$.

Step 0. Set $k = 0$.

Step 1. Set $i = 1$, $y_k^1 = x_k$.

(continued)

Step 2. Compute the maximum feasible step-size $\bar{\alpha}_k^i$ along d_i .

Set $\alpha_k^i = \min\{\tilde{\alpha}_k^i, \bar{\alpha}_k^i\}$

If $\alpha_k^i > 0$ and $f(y_k^i + \alpha_k^i d_i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2$, then

perform the *expansion step*:

While $f(y_k^i + \delta\alpha_k^i d_i) \leq f(y_k^i) - \gamma(\delta\alpha_k^i)^2$ and $\delta\alpha_k^i \leq \bar{\alpha}_k^i$

set $\alpha_k^i = \delta\alpha_k^i$;

End While

set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$;

Else set $\alpha_k^i = 0$, $\tilde{\alpha}_{k+1}^i = \theta\tilde{\alpha}_k^i$.

End If

Step 3. Set $y_k^{i+1} = y_k^i + \alpha_k^i d_i$.

Step 4. If $i \leq 2n$ set $i = i + 1$ and go to Step 2.

Step 5. Compute $x_{k+1} \in S$ such that $f(x_{k+1}) \leq f(y_k^{2n+1})$, set $k = k + 1$,

and go to Step 1.

Remark 20.1 Assume $\bar{\alpha}_k^i \geq \tilde{\alpha}_k^i > 0$. Then by the instructions of the algorithm we have that only one of the following conditions hold:

- (I) $\alpha_k^i = 0$ and $f(y_k^i + \tilde{\alpha}_k^i d_i) \geq f(y_k^i) - \gamma(\tilde{\alpha}_k^i)^2$;
- (II) $0 < \alpha_k^i < \bar{\alpha}_k^i$ and $\delta\alpha_k^i \leq \bar{\alpha}_k^i$ and

$$f(y_k^i + \alpha_k^i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2 \quad \text{and} \quad f(y_k^i + \delta\alpha_k^i) > f(y_k^i) - \gamma(\delta\alpha_k^i)^2;$$

- (III) $0 < \alpha_k^i \leq \bar{\alpha}_k^i$ and $\delta\alpha_k^i > \bar{\alpha}_k^i$ and

$$f(y_k^i + \alpha_k^i) \leq f(y_k^i) - \gamma(\alpha_k^i)^2.$$

□

Remark 20.2 Steps 1–4 produce the points $y_k^1 = x_k, y_k^2, \dots, y_k^{2n}, y_k^{2n+1}$ by sampling the $2n$ directions. At Step 5 it is possible to choose any point x_{k+1} such that $f(x_{k+1}) \leq f(y_k^{2n+1})$. In particular, we can set $x_{k+1} = y_k^{2n+1}$. \square

Preliminarily we state the following result.

Proposition 20.12 *Let $f : R^n \rightarrow R$ be a continuous function. Then:*

- (i) *the expansion procedure is well-defined, i.e., the while cycle terminates in a finite number of inner iterations;*
- (ii) *denoting by $\{\alpha_k^i\}$, $\{\tilde{\alpha}_k^i\}$, for $i = 1, \dots, 2n$, the sequences of scalars produced by MC-Box, we have*

$$\lim_{k \rightarrow \infty} \alpha_k^i = 0 \quad (20.28)$$

$$\lim_{k \rightarrow \infty} \tilde{\alpha}_k^i = 0. \quad (20.29)$$

Proof In order to prove (i), assume by contradiction that the while cycle does not terminate at iteration k for some direction d_i . Then we have for $j = 0, 1, \dots$

$$f(y_k^i + \delta^j \tilde{\alpha}_k^i d_i) \leq f(y_k^i) - \gamma (\delta^j \tilde{\alpha}_k^i)^2, \quad (20.30)$$

where the points $y_k^i + \delta^j \tilde{\alpha}_k^i d_i$ are feasible points. Taking the limits for $j \rightarrow \infty$ we obtain that $f(y_k^i + \delta^j \tilde{\alpha}_k^i d_i) \rightarrow -\infty$, and this contradicts the fact that f is bounded below over S .

Let us prove assertion (ii). The instructions of the algorithm imply that for all k we have $x_k \in S$ and $f(x_{k+1}) \leq f(x_k)$, and hence, as f is continuous and hence bounded below over the compact set S , it follows that the sequence $\{f(x_k)\}$ converges. Furthermore we can write

$$f(x_{k+1}) \leq f(x_k) - \gamma \sum_{i=1}^{2n} (\alpha_k^i)^2,$$

from which, taking into account the convergence of $\{f(x_k)\}$, it follows that (20.28) holds.

In order to prove (20.29), for any $i \in \{1, \dots, 2n\}$ we split the set of indices of iterates $\{0, 1, \dots\}$ into two subsets K and \bar{K} (for notational convenience we omit the dependence on i), where:

- $k \in K$ if and only if $\alpha_k^i > 0$;
- $k \in \bar{K}$ if and only if $\alpha_k^i = 0$.

For each $k \in K$ we have $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, and hence, if K is an infinite subset, from (20.28) it follows

$$\lim_{k \in K, k \rightarrow \infty} \tilde{\alpha}_{k+1}^i = 0. \quad (20.31)$$

For every $k \in \bar{K}$, let $m_k < k$ be the biggest index such that $m_k \in K$ (we assume $m_k = 0$ if this index does not exist, i.e., K is empty). We can write

$$\tilde{\alpha}_{k+1}^i = (\theta)^{k+1-m_k} \tilde{\alpha}_{m_k}^i \leq \tilde{\alpha}_{m_k}^i$$

For $k \in \bar{K}$ and $k \rightarrow \infty$ we have that either $m_k \rightarrow \infty$ (if K has an infinite subset) or $k - m_k \rightarrow \infty$ (if K is finite). Then (20.31) and the fact that $\theta \in (0, 1)$ imply $\tilde{\alpha}_{k+1}^i \rightarrow 0$ for $k \in \bar{K}$ and $k \rightarrow \infty$. \square

We can state the following convergence result.

Proposition 20.13 *Let $f : R^n \rightarrow R$ be a continuously differentiable function and let $\{x_k\}$ be the sequence generated by MC-Box. Then every limit point of $\{x_k\}$ is a critical point.*

Proof By contradiction let us assume that there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{k \in K, k \rightarrow \infty} x_k = \bar{x},$$

and \bar{x} is not a critical point, i.e.,

$$\nabla f(\bar{x})^T \bar{d} < 0, \quad (20.32)$$

where \bar{d} is a feasible direction at \bar{x} . From (20.25) it follows that there exists an index set $J \subseteq \{1, \dots, 2n\}$ such that

$$\bar{d} = \sum_{j \in J} \beta_j d_j,$$

where $\beta_j \geq 0$ and $d_j \in D \cap D(\bar{x})$ for all $j \in J$. Then, using (20.32) we have that

$$\nabla f(\bar{x})^T d_j < 0 \quad (20.33)$$

for some $j \in J$. For all $k \in K$ we have

$$\|y_k^j - x_k\| \leq \sum_{i=1}^{j-1} \alpha_k^i,$$

so that, recalling (20.28), we obtain

$$\lim_{k \in K, k \rightarrow \infty} y_k^j = \bar{x}. \quad (20.34)$$

From Proposition 20.10 we get that for $k \in K$ and k sufficiently large $\bar{\alpha}_k^j \geq \bar{\alpha}$ for some $\bar{\alpha} > 0$, where $\bar{\alpha}_k^j$ is the maximum feasible step size along d_j starting from x_k . Then, using Proposition 20.12, for $k \in K$ and k sufficiently large we have $\bar{\alpha}_k^j \geq \tilde{\alpha}_k^j$ and that either (I) or (II) of Remark 20.1 holds (note that (III) can not hold since $\alpha_k^j \rightarrow 0$ and $\bar{\alpha}_k^j \geq \bar{\alpha} > 0$). This means that either $\alpha_k^j = 0$ and

$$f(y_k^j + \tilde{\alpha}_k^j d_j) \geq f(y_k^j) - \gamma (\tilde{\alpha}_k^j)^2,$$

or $\alpha_k^j > 0$ and

$$f(y_k^j + \delta \alpha_k^j d_j) \geq f(y_k^j) - \gamma (\delta \alpha_k^j)^2.$$

Now, for each $k \in K$ and k sufficiently we set

$$\eta_k^j = \begin{cases} \delta \alpha_k^j & \text{if } \alpha_k^j > 0 \\ \tilde{\alpha}_k^j & \text{otherwise,} \end{cases}$$

and hence we can write

$$f(y_k^j + \eta_k^j d_j) \geq f(y_k^j) - \gamma (\eta_k^j)^2. \quad (20.35)$$

By the mean value theorem we have

$$\nabla f(\xi_k^j)^T d_j \geq -\gamma \eta_k^j, \quad (20.36)$$

where $\xi_k^j = y_k^j + t_k \eta_k^j d_j$ and $t_k \in (0, 1)$. From (20.34), as $\eta_k^j \rightarrow 0$ for $k \in K$ and $k \rightarrow \infty$, it follows that $\xi_k^j \rightarrow \bar{x}$. Then, taking the limits for $k \in K$ and $k \rightarrow \infty$

in (20.36) and recalling the continuity of the gradient we obtain

$$\nabla f(\bar{x})^T d_j \geq 0,$$

which contradicts (20.33). \square

20.6 Concave Programming for Minimizing the Zero-Norm Over Polyhedral Sets

Let us consider the problem

$$\begin{aligned} & \min_{x \in R^n} \|x\|_0 \\ & x \in P \end{aligned} \tag{20.37}$$

where $P \subset R^n$ is a polyhedral set, $\|x\|_0$ is the so-called *zero-norm* defined as

$$\|x\|_0 = \text{card}\{i : x_i \neq 0, i = 1, \dots, n\},$$

that is, the zero-norm of a vector is the number of its nonzero components. Note that the zero-norm is not a norm.

This sparse optimization problem is an NP-Hard combinatorial optimization problem [4], and arises in various fields such as machine learning, pattern recognition, signal processing.

Problem (20.37) can be formulated as a *Mixed Integer Linear Programming* (MILP) problem, and several heuristic techniques have been proposed in the literature for solving it. Here we limit ourselves to analyze the approach based on concave programming.

To this aim, by introducing the *step function* $s : R^+ \rightarrow \{0, 1\}$ such that $s(t) = 1$ if $t \neq 0$ and $s(t) = 0$ if $t = 0$, problem (20.37) can be equivalently rewritten as follows

$$\min_{x \in R^n} \sum_{i=1}^n s(|x_i|) \tag{20.38}$$

$$x \in P.$$

The objective function of problem (20.38) is discontinuous. The idea underlying the approach based on concave programming is that of approximating the discontinuous function $s(t)$ by a continuously differentiable, concave function. For instance, the

following function

$$f(t) = 1 - e^{-\alpha t}, \quad (20.39)$$

with $\alpha > 0$, is such that

$$\lim_{\alpha \rightarrow \infty} f(t) = \begin{cases} 1 & \text{if } t > 0 \\ 0 & \text{if } t = 0 \end{cases} \quad (20.40)$$

Then, function (20.39) can be considered a suitable approximation of $s(t)$. We replace problem (20.38) by the following problem

$$\min_{x \in R^n} \sum_{i=1}^n f(|x_i|) \quad (20.41)$$

$$x \in P.$$

We observe that function f is monotone increasing in R^+ , so that, problem (20.41) is equivalent (see the chapter on the equivalence between problems) to the following problem

$$\min_{x \in R^n, y \in R^n} \sum_{i=1}^n (1 - e^{-\alpha y_i}) \quad (20.42)$$

$$x \in P$$

$$-y_i \leq x_i \leq y_i \quad i = 1, \dots, n.$$

Remark 20.3 A well-known result states that a concave function, bounded below on a nonempty polyhedral set admits minimum on it. Then problem (20.42) admits solution. \square

We state the following assumption.

Assumption 20.1 *The polyhedral set P admits at least a vertex.* \square

The above assumption implies that the polyhedral set defining the feasible set of problem (20.42), that is, the polyhedral set

$$\mathcal{Q} = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} : x \in P, -y_i \leq x_i \leq y_i \quad i = 1, \dots, n \right\},$$

admits a vertex.

Remark 20.4 From known results we get that problem (20.42) admits an optimal solution which is a vertex solution. Indeed, it can be proved that a concave function, bounded below on an nonempty polyhedral set having at least a vertex, admits an optimal solution which is a vertex solution (see, e.g., [247]). \square

The original problem (20.38) can be rewritten in the equivalent form involving the variables x, y :

$$\begin{aligned} \min_{x \in R^n, y \in R^n} & \sum_{i=1}^n s(y_i) = \|y\|_0 \\ x \in P \\ -y_i \leq x_i \leq y_i \quad i = 1, \dots, n. \end{aligned} \tag{20.43}$$

We can prove the following equivalence result.

Proposition 20.14 *There exists a value $\bar{\alpha} > 0$ such that for any $\alpha \geq \bar{\alpha}$ problem (20.42) has a vertex solution which is also solution of the original problem (20.43).*

Proof Assume, by contradiction, that there exists an infinite sequence $\{\alpha^k\}$, with $\alpha^k \rightarrow \infty$, such that any vertex solution of (20.42) is not solution of (20.43).

Let $z^* = (x^*, y^*)$ be a solution of (20.43), and let $z^k = (x^k, y^k)$ be a vertex solution of (20.42) with $\alpha = \alpha^k$. The number of vertices is finite, so that there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that $z^k = \bar{z}$ for every $k \in K$. Then we have

$$\|y^*\|_0 < \|\bar{y}\|_0. \tag{20.44}$$

As (\bar{x}, \bar{y}) is a vertex solution of (20.42) with $\alpha = \alpha^k$ and $k \in K$, we can write

$$\sum_{i=1}^n \left(1 - e^{-\alpha^k \bar{y}_i}\right) \leq \sum_{i=1}^n \left(1 - e^{-\alpha^k y_i^*}\right).$$

Taking the limits for $k \in K$ and $k \rightarrow \infty$, and recalling (20.40) we obtain

$$\sum_{i=1}^n (s(|\bar{y}_i|)) = \|\bar{y}\|_0 \leq \sum_{i=1}^n (s(|y_i^*|)) = \|y^*\|_0,$$

and this contradicts (20.44). \square

A similar concave optimization-based approach is that using the logarithm function instead of the step function, and this leads to a concave smooth problem of the form

$$\begin{aligned} \min_{x \in R^n, y \in R^n} & \sum_{i=1}^n \ln(y_i + \epsilon) \\ x \in P & \\ -y_i \leq x_i \leq y_i & \quad i = 1, \dots, n. \end{aligned} \tag{20.45}$$

with $0 < \epsilon \ll 1$. Formulation (20.45) is practically motivated by the fact that, due to the form of the logarithm function, it is better to increase one variable y_i while setting to zero another one rather than doing some compromise between both, and this should facilitate the computation of a sparse solution. We can state the following result (see, e.g., [225]) showing the equivalence between the original problem (20.43) and the smooth concave problem (20.45).

Proposition 20.15 Assume that problem (20.43) admits a solution y^* such that $\|y^*\|_0 < n$. There exists a value $\bar{\epsilon} > 0$ such that, for any $\epsilon \in (0, \bar{\epsilon}]$, problem (20.45) has a vertex solution which is also solution of the original problem (20.43).

As shown in the next section, the Frank-Wolfe method converges in a finite number of iterations to a critical point of a problem of the form (20.42) (or (20.45)). Therefore, it can be a suitable method, coupled with a global optimization strategy, for solving zero-norm problems employing the described concave programming approach.

20.7 Frank-Wolfe Method Applied to Concave Programming Problems

Let us consider the problem

$$\begin{aligned} \min f(x) \\ x \in P \end{aligned} \tag{20.46}$$

where $f : R^n \rightarrow R$ is a concave, continuously differentiable function, bounded below on the polyhedral set P . We assume that there exists at least a vertex of P . Note that problems (20.42) and (20.45) belong to the considered class of concave programming problems. We show that the Frank-Wolfe method with unitary stepsize, i.e., without line search, converges in a finite number of iterations to a critical point of (20.46).

We formally describe the method.

Method of Frank-Wolfe with Unitary Stepsize (FW1)

1. Choose the initial point $x_0 \in P$.
- For k=0,1,...
2. Compute a vertex solution \hat{x}_k of problem

$$\min_{x \in P} \nabla f(x_k)^T x \quad (20.47)$$

- If $\nabla f(x_k)^T (\hat{x}^k - x_k) = 0$ stop.
3. Set $x_{k+1} = \hat{x}_k$.
- End For

We can state the following finite convergence result.

Proposition 20.16 *The Frank-Wolfe algorithm with unitary stepsize is well-defined and converges to a critical point of problem (20.46) in a finite number of iterations.*

Proof In order to show that the method is well-defined, we must prove that the linear programming problem (20.47) is bounded below so that, being P not empty, it admits a vertex solution. From the boundedness and the concavity of f it follows

$$-\infty < \inf_{x \in P} f(x) - f(x_k) \leq f(x) - f(x_k) \leq \nabla f(x_k)^T (x - x_k),$$

and we can conclude the linear programming problem (20.47) admits a vertex solution \hat{x}_k . If x_k is not a critical point then we have

$$\nabla f(x_k)^T (\hat{x}_k - x_k) < 0.$$

Therefore, recalling the concavity of f , we can write

$$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) = f(x_k) + \nabla f(x_k)^T (\hat{x}_k - x_k) < f(x_k).$$

Then, as the points x_1, x_2, \dots generated by the method are vertices of P and the number of vertices of P is finite, using the fact that the sequence of function values is strictly decreasing, we can conclude that the algorithm stops in a finite number of iterations at a point x_k such that

$$\nabla f(x_k)^T (\hat{x}_k - x_k) = 0,$$

which implies

$$\nabla f(x_k)^T (x - x_k) \geq 0 \quad \text{for every } x \in P,$$

i.e., that x_k is a critical point.

□

20.8 Exercises

20.1 Define a computer code based on the Frank-Wolfe method, for minimizing a quadratic convex function with box constraints and perform some numerical experiments.

20.2 Define a computer code based on the gradient projection method for minimizing a continuously differentiable function with box constraints and perform some numerical experiments.

20.3 Define a computer code based on the Frank-Wolfe method (or on the gradient projection method) for minimizing a quadratic convex function with simplex constraints and perform some numerical experiments.

20.9 Notes and References

The literature on the gradient projection method is wide (see, e.g., [16] as one of the main references). Later we will briefly analyze nonmonotone gradient projection methods based on the Barzilai-Borwein stepsize. The Frank-Wolfe was originally proposed in [100] for the solution of quadratic problems. The method has been widely used, for instance, in the context of network equilibrium problems [99], that are typically large-scale problems. Furthermore, it is advantageously employed in the solution of large-scale problems with concave objective functions and feasible

sets defined by polyedra (see, e.g., [180, 225, 255]). A survey on Frank-Wolfe method can be found in [27]. The approach of sparse optimization based on concave programming has been proposed in [180]. The derivative-free method for box constrained problems has been presented in [174].

Chapter 21

Penalty and Augmented Lagrangian Methods



In this chapter we consider an important class of methods for the analysis and the solution of constrained optimization problems, based on the construction of a (finite or infinite) sequence of unconstrained problems. After a short introduction, we illustrate the essential theoretical properties of penalty and augmented Lagrangian methods and we describe some basic computational approaches.

21.1 Basic Concepts

Penalty and augmented Lagrangian methods are typically *exterior methods* that may produce points in the exterior of the feasible set. These methods consist, essentially, in the introduction of a *merit function*, such that, under appropriate assumptions, a finite or infinite sequence of unconstrained minimizations of this function yields (or approximates) a solution of the constrained problem. The merit function has the role of balancing the reduction of the objective with the improvement in constraint satisfaction and depends on parameters (and possibly on auxiliary variables, such as KKT multiplier estimates), which are updated during the solution process, so that a sequence of subproblems is generated. To introduce these techniques, let us consider the constrained problem:

$$\min_{x \in S} f(x),$$

where $S \subset R^n$ is a nonempty closed subset of R^n and $f : R^n \rightarrow R$. Suppose we construct a function $\psi : R^n \rightarrow R^+$ such that $\psi(x) = 0$ for $x \in S$ and $\psi(x) > 0$ for

$x \notin S$. We can define a merit function P , which we call *penalty function*¹ by letting

$$P(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \psi(x),$$

where $\varepsilon > 0$ is a positive scalar called *penalty parameter*.

Then we can attempt to solve the unconstrained problem of minimizing $P(\cdot, \varepsilon)$ on R^n , for given values of ε . It is easily seen that, when x is infeasible and ε is sufficiently small, the minimization of P places large penalties on feasibility violation, whereas when x is feasible, only the original objective function is minimized. Thus, by reducing ε when required, we can attempt to reach a constrained minimizer through an unconstrained minimization process. This idea is essentially at the basis of various merit functions proposed in constrained optimization, which have quite different structures and that can be classified from different points of view.

A first distinction, related to the variables of the merit function, is that between *penalty functions*, which depend only on the problem variables, and *augmented Lagrangian functions*, which depend both on the problem variables and on some estimates of the KKT multipliers. Augmented Lagrangian functions are typically obtained by adding penalty terms to the ordinary Lagrangian function and the criterion used for choosing and updating the multiplier estimates may correspond to various different techniques.

A second important distinction, at least from a conceptual point of view, is related to the notion of *exactness*. We can distinguish, in particular, between *exact* and *sequential* penalty functions. Typically, we speak of an exact penalty function when there exists a threshold value ε^* of a penalty parameter such that, for $0 < \varepsilon \leq \varepsilon^*$, a single unconstrained minimization of the merit function yields the required correspondence with the constrained problem. Sequential penalty functions are those such that a positive threshold value for the penalty parameter does not exist and the correspondence with the constrained problem can only be established in the limit for $\varepsilon \rightarrow 0$, through the solution of a sequence of unconstrained problems.

Actually, the precise notion of “exactness” is not the same for all authors. Here, following [75], but simplifying the terminology, we will refer, informally, to a notion motivated essentially by the intended computational use of a penalty function. More specifically, we say that a penalty function $P(\cdot; \varepsilon)$ is exact if there exists a value ε^* of the penalty parameter such that, for each $0 < \varepsilon \leq \varepsilon^*$, we have:

- (i) every global minimum point of the constrained problem is also a global minimum point of $P(\cdot; \varepsilon)$ and conversely;
- (ii) every local minimizer of $P(\cdot; \varepsilon)$ is also a local minimizer of the constrained problem;
- (iii) every critical point of $P(\cdot; \varepsilon)$ is also a critical point of the constrained problem and conversely.

¹ Note that in many works the term penalty function is referred to the penalty term ψ .

We note that, in general, the converse of (ii) may be not required to give a useful meaning to the notion of exactness, as property (i) already guarantees that no global solution is missed and property (iii) ensures that critical points are preserved. However, also the converse of (ii) can be established under suitable conditions (see, e.g. [75]).

Property (iii) is at the basis of properties (i) and (ii) and it is of special interest in non convex problems. In these cases, the *local* techniques considered in this book typically guarantee, as already discussed, that constrained and unconstrained methods determine, in principle, only *good* critical points, that is points satisfying necessary optimality conditions, which yield an improvement with respect to non critical starting points. Thus the correspondence of critical points is quite important from a computational point of view.

A similar concept of exactness can be referred to augmented Lagrangian functions. We speak of *exact augmented Lagrangian* when the solutions of the constrained problem can be obtained through a single unconstrained minimization of the merit function, with respect to both problem variables and multipliers estimates.

Under the assumption that all problem functions are continuously differentiable, exact penalty functions can be distinguished, in turn, into *non differentiable* and *continuously differentiable* exact penalty functions, on the basis of the structure of the penalty term. Many other distinctions can be introduced in relation to the structure of the constrained problem and to the specific choice of the penalty term.

In all penalty and augmented Lagrangian methods an (often tacit) assumption is that the sequences of points generated by the solution algorithms have limit points in the interior of a *compact set*. Actually, there are specific motivations for compactness, in dependence of the problem features, the structure of the merit function and the solution algorithms.

In order to simplify our discussion, we refer to a solution algorithm organized as a sequence of (possibly approximate) unconstrained minimizations that produce the points x_k , in correspondence to each penalty parameter ε_k .

A first requirement is that these steps are well defined and a sufficient condition is that, for each k , the level set of $P_k \equiv P(\cdot; k)$, that is, the set

$$\mathcal{L}_{P_k} = \{x \in R^n : P(x; \varepsilon_k) \leq P(\bar{x}_k, \varepsilon_k)\}$$

is non empty and compact for every fixed \bar{x}_k . Under usual smoothness assumptions, this in turn implies that an unconstrained critical point of P_k can be determined in a finite number of *local inner steps* with any desired precision.

The second important requirement concerns a compactness condition on the sequence of major iterations formed by the points x_k for decreasing values of ε_k . We must show that this sequence has limit points and that all these points are critical points of the constrained problem. As we will see, in the general unstructured case, finding *a priori conditions* that guarantee both the *existence* of limit points and the correspondence with problem solutions can be quite difficult. We note, in particular, that these conditions must guarantee that the feasible set is non empty.

When a feasible point is not easily available, ensuring feasibility, as we know, is a problem of global nature. Thus, the conditions to be imposed on the region where the unconstrained minimization of the merit function is performed should include both compactness and regularity conditions, in order to establish constructively, through the adoption of *local* minimization techniques, that the feasible set is non empty.

Compactness assumptions, as we will see, are also essential for establishing the existence of a non zero threshold value of the penalty parameter in exact penalty methods.

The need of imposing compactness, without creating spurious critical points or minimizers, has motivated also the definition of *shifted barrier terms*, to be included in the merit function. Some of these techniques will be described in the sequel, in connection with exact penalty methods.

The results on unconstrained merit functions mentioned above can also be extended to the constrained minimization of penalty functions or augmented Lagrangian functions, subject only to some of the original constraints, such as, for instance, nonnegativity constraints or simple bounds on the variables. In the sequel, however, we will be essentially concerned with the unconstrained case and we will refer to the literature for these extensions.

The computational use of penalty and augmented Lagrangian methods is a point of major concern in the field of nonlinear optimization. In principle, the merit function can be minimized by employing the well known algorithms for unconstrained minimization for each parameter choice, but, more generally, the merit function can be used for globalizing some local algorithm. In this case, starting from some given initial point, possibly infeasible, a new tentative point is generated at each major step, by solving a local subproblem (typically constrained) and the merit function is used for evaluating and correcting, if needed, the tentative point.

In the next sections we will describe the essential conceptual features of sequential penalty methods, augmented Lagrangian methods, exact penalty methods and exact augmented Lagrangian methods. Computational applications of penalty and augmented Lagrangian functions will be also considered in the next chapter devoted to sequential quadratic programming problems.

21.2 Sequential Penalty Functions

The simplest (and probably oldest) techniques employing an unconstrained merit function are the sequential penalty functions, whose general structure has been introduced in the preceding section.

Consider again the problem

$$\min_{x \in S} f(x) \quad (21.1)$$

and the merit function

$$P(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \psi(x),$$

where $\psi : R^n \rightarrow R^+$ is a continuous function such that

$$\psi(x) \begin{cases} = 0, & \text{if } x \in S \\ > 0 & \text{if } x \notin S. \end{cases} \quad (21.2)$$

We can establish the following result, where we assume that $\{x_k\}$ is a sequence of points in R^n , generated starting from a given initial point $x_0 \in R^n$, in correspondence to a sequence of penalty parameters $\{\varepsilon_k\}$.

Proposition 21.1 (Convergence of Sequential Penalty Functions) *Let S be a non empty closed subset of R^n , let $f : R^n \rightarrow R$ be continuous on R^n and let $\{\varepsilon_k\}$ be a sequence of positive numbers such that $\varepsilon_k \rightarrow 0$. Suppose that*

- (i) *problem (21.1) has an optimal solution, that is, there exists $x^* \in S$ such that*

$$f(x^*) = \min_{x \in S} f(x);$$

- (ii) *for every $\varepsilon_k > 0$ there exists $x_k \in R^n$ such that*

$$P(x_k; \varepsilon_k) = \min_{x \in R^n} P(x; \varepsilon_k);$$

- (iii) *there exists a compact set $D \subset R^n$ such that $x_k \in D$ for all k .*

Then, the sequence $\{x_k\}$ has limit points and every limit point is an optimal solution of Problem (21.1).

Proof Let x^* be a global minimum point of problem (21.1), so that $x^* \in S$ and $\psi(x^*) = 0$. As x_k is a global minimum point of $P(x_k; \varepsilon_k)$, we have that

$$f(x^*) = P(x^*; \varepsilon_k) \geq P(x_k; \varepsilon_k) = f(x_k) + 1/\varepsilon_k \psi(x_k),$$

which implies that

$$f(x^*) \geq f(x_k) \quad (21.3)$$

and that

$$\varepsilon_k (f(x^*) - f(x_k)) \geq \psi(x_k). \quad (21.4)$$

As D is compact the sequence $\{x_k\}$ has limit points in D . Let $\tilde{x} \in D$ be a limit point and let $\{x_k\}_K$ be a subsequence converging to \tilde{x} . Then, from (21.4), taking

limits for $k \in K$ it follows, by the continuity of ψ , that $\psi(\tilde{x}) = 0$ and hence that $\tilde{x} \in S$, so that $f(\tilde{x}) \geq f(x^*)$. On the other hand, taking limits in (21.3) in the same subsequence, we obtain $f(x^*) \geq f(\tilde{x})$, and hence we have $f(x^*) \geq f(\tilde{x}) \geq f(x^*)$, which implies $f(\tilde{x}) = f(x^*)$ and this completes the proof. \square

We note that the assumptions stated in the preceding simple proposition imply several restrictions on the properties of the original constrained problem and on the definition of the penalty term. Assumption (i) requires that the feasible set is non empty and that there exists an optimal solution of the constrained problem.

Assumption (ii) requires that the unconstrained problem has an optimal solution. This can be guaranteed, for instance, if $P(\cdot, \varepsilon_k)$ is coercive for each k . If we assume that $\varepsilon_{k+1} \leq \varepsilon_k$ we have that $P(x; \varepsilon_k) \geq P(x; \varepsilon_1)$ for every k and $x \in R^n$ and hence, if $P(\cdot; \varepsilon_1)$ is coercive, condition (ii) holds for every k .

As we know, the compactness condition (iii) is another crucial assumption and it is equivalent to the *a posteriori* assumption that every subsequence has a limit point. If we suppose that there exists $x_0 \in S$ such that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact, then (i) is obviously satisfied and also (iii) holds if (ii) is valid. In fact, recalling the proof of the preceding proposition, from (21.3) we have that

$$f(x_k) \leq f(x^*) \leq f(x_0)$$

and hence that $x_k \in \mathcal{L}_0$.

There are various penalty functions that show the structure considered above. One of the best known functions is the *quadratic penalty function*, which can be introduced for solving a constrained problem of the form:

$$\begin{aligned} \min & f(x) \\ g(x) &\leq 0, \\ h(x) &= 0, \end{aligned} \tag{21.5}$$

where $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$, $h : R^n \rightarrow R^p$ are continuously differentiable.

The quadratic penalty function for this problem is the function

$$P(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \left(\sum_{i=1}^m (g_i^+(x))^2 + \sum_{i=1}^p h_i^2(x) \right), \tag{21.6}$$

where $g_i^+(x) = \max\{g_i(x), 0\}$. The function P is continuously differentiable on R^n with gradient

$$\nabla P(x; \varepsilon) = \nabla f(x) + \frac{2}{\varepsilon} \left(\sum_{i=1}^m g_i^+(x) \nabla g_i(x) + \sum_{i=1}^p h_i(x) \nabla h_i(x) \right).$$

We note that, in general, P is not twice continuously differentiable, since the function defined by $\max\{g_i(x), 0\}$ could not be differentiable at points where $g_i(x) = 0$. However, we could consider penalty terms of the form $\max^q\{g_i(x), 0\}$ for $q \geq 3$, in order to obtain a twice continuously differentiable function. Here, for simplicity, we will refer, when required, to the quadratic penalty function and the possible extensions are left to the reader.

From a computational point of view, the minimization of P by means of standard algorithms for unconstrained minimization can be difficult, because of the need of employing large values of $1/\varepsilon$ (ideally $\varepsilon \rightarrow 0$), which can determine a severe ill conditioning of the Hessian matrix $\nabla^2 P$ (at points where it is defined)). Thus, it could be convenient to solve approximately a sequence of unconstrained problems, by reducing gradually ε_k in a way that the minimization of $P(\cdot, \varepsilon_{k+1})$ is effected starting from the (approximate) minimizer x_k of $P(\cdot, \varepsilon_k)$ (or from a better point), which should not be too far from x_{k+1} , if the penalty parameter has been reduced slowly.

In the general, non convex case, we can define a computational scheme where, at each k , the unconstrained minimization of P is performed using an efficient unconstrained algorithm for continuously differentiable problems.

In order to define an algorithm model, we first introduce the following assumption, which yields a sufficient condition for guaranteeing that an approximate stationary point of P can be computed at each step.

Assumption 21.1 *There exists $\varepsilon_1 > 0$ such that, for each $0 < \varepsilon \leq \varepsilon_1$ the function $P(\cdot; \varepsilon)$ is continuous and coercive on R^n .* □

An immediate consequence of the preceding assumption is that, if $\bar{x} \in R^n$ and $0 < \varepsilon \leq \varepsilon_1$ then the level set

$$\mathcal{L}_{P_\varepsilon} = \{x \in R^n : P(x; \varepsilon) \leq P(\bar{x}, \varepsilon)\}$$

is non empty and compact. Therefore, when all the problem functions are continuously differentiable, the unconstrained algorithms considered in the preceding chapters can determine, in principle, an approximate stationary point of P with the required precision.

In the next scheme we will assume that the starting point at each step k is chosen as the point, say \hat{x}_k , that yields the minimum value, between $P(x_0; \varepsilon_k)$ and $P(x_{k-1}; \varepsilon_k)$. As we will see in the sequel, the motivation could be that of starting from a feasible point x_0 , if available. Thus at each step we will determine the approximate critical point x_k in a way that

$$P(x_k; \varepsilon_k) \leq \min\{P(x_0; \varepsilon_k), P(x_{k-1}; \varepsilon_k)\},$$

starting from \hat{x}_k . Then, at least under these assumptions, the following conceptual scheme (where we omit any practical termination criterion) is well defined.

Algorithm 21.1 (Sequential Penalty Algorithm)

Data $x_0 \in R^n$, $\varepsilon_1 > 0$, $\theta \in (0, 1)$ and a sequence $\{\xi_k\}$ such that $\xi_k > 0$ and $\lim_{k \rightarrow \infty} \xi_k = 0$,
For k=1, 2, ...

1. Using an unconstrained minimization algorithm, determine x_k such that
 $\|\nabla P(x_k; \varepsilon_k)\| \leq \xi_k$ and

$$P(x_k; \varepsilon_k) \leq \min\{P(x_0; \varepsilon_k), P(x_{k-1}; \varepsilon_k)\}$$

2. Set $\varepsilon_{k+1} = \theta \varepsilon_k$.

End for

Under suitable assumptions, we can show that the algorithm converges towards KKT points of the constrained problem (21.5).

Proposition 21.2 (Convergence to KKT Points) *Let f, g, h be continuously differentiable on R^n , let x_0 be a given point in R^n , and let P be the function defined in (21.6). For $k = 1, 2, \dots$, let $\{\varepsilon_k\}$ and $\{\xi_k\}$ be sequences of positive numbers such that $\lim_{k \rightarrow \infty} \xi_k = 0$, $\lim_{k \rightarrow \infty} \varepsilon_k = 0$. Assume that:*

- (i) *for every k , we can compute $x_k \in R^n$ such that: $\|\nabla P(x_k; \varepsilon_k)\| \leq \xi_k$; and that*

$$P(x_k; \varepsilon_k) \leq \min\{P(x_0; \varepsilon_k), P(x_{k-1}; \varepsilon_k)\}$$

- (ii) *there exists an infinite subsequence $\{x_k\}_K$ such that, for every $k \in K$, the point x_k belongs to a compact set $D \subset R^n$;*
- (iii) *for every $x \in D$ the gradients*

$$\{\nabla h_i(x), i = 1, \dots, p, \quad \nabla g_i(x), i \in I_+(x)\},$$

where $I_+(x) = \{i : g_i(x) \geq 0\}$ are linearly independent.

Then, if $x^ \in D$ is a limit point of $\{x_k\}_K$, the point x^* is feasible and there exist λ^*, μ^* such that (x^*, λ^*, μ^*) satisfies the Karush-Kuhn-Tucker conditions for problem (21.5).*

Proof By assumption, there exists a subsequence $\{x_k\}_K$ of points produced by the algorithm, which belong to the compact set D . Then we can extract another subsequence, (which we relabel, $\{x_k\}$) converging to some $x^* \in D$. Recalling the expression of ∇P , letting $\lambda_k = 2/\varepsilon_k g^+(x_k)$ and $\mu_k = 2/\varepsilon_k h(x_k)$, where $g^+(x_k)$ is the vector with components $g_i^+(x_k) = \max\{g_i(x_k), 0\}$, we can write

$$\nabla P(x_k; \varepsilon_k) = \nabla f(x_k) + \nabla g(x_k)\lambda_k + \nabla h(x_k)\mu_k.$$

As x_k converges to x^* , by continuity of g , for sufficiently large values of k , we have $g_i^+(x_k) = 0$ for every $i \notin I_+(x^*)$ and hence we have $(\lambda_k)_i = 0$ for every $i \notin I_+(x^*)$. Now, denoting by $\tilde{\lambda}_k$ the vector with components $(\tilde{\lambda}_k)_i = 2/\varepsilon_k g_i^+(x_k)$, $i \in I_+(x^*)$, and by $\nabla g_{I_+(x^*)}(x_k)$ the matrix with columns $\nabla g_i(x_k)$, for $i \in I_+(x^*)$, we can write

$$\nabla P(x_k; \varepsilon_k) = \nabla f(x_k) + \nabla g_{I_+(x^*)}(x_k)\tilde{\lambda}_k + \nabla h(x_k)\mu_k. \quad (21.7)$$

Letting

$$M(x_k) = (\nabla g_{I_+(x^*)}(x_k) \ \nabla h(x_k)) \quad u_k = \begin{pmatrix} \tilde{\lambda}_k \\ \mu_k \end{pmatrix},$$

Eq. (21.7) can be rewritten in the form:

$$\nabla P(x_k; \varepsilon_k) = \nabla f(x_k) + M(x_k)u_k. \quad (21.8)$$

By assumption (iii), for sufficiently large k , the matrix M has linearly independent columns and hence the matrix $M(x_k)^T M(x_k)$ is non singular. Therefore, from (21.8), we get:

$$u_k = [M(x_k)^T M(x_k)]^{-1} M(x_k)^T (\nabla P(x_k; \varepsilon_k) - \nabla f(x_k)) \quad (21.9)$$

Now, by assumption, the gradient $\nabla P(x_k; \varepsilon_k)$ converges to 0, and hence we get

$$\lim_{k \rightarrow \infty} u_k = u^* = -[M(x^*)^T M(x^*)]^{-1} M(x^*)^T \nabla f(x^*),$$

where:

$$u^* = \begin{bmatrix} \lambda_{I_+(x^*)}^* \\ \mu^* \end{bmatrix} = \lim_{k \rightarrow \infty} \begin{bmatrix} \tilde{\lambda}_k \\ \mu_k \end{bmatrix}.$$

As a consequence, from (21.8) we obtain, in the limit

$$\nabla f(x^*) + M(x^*)u^* = 0, \quad (21.10)$$

whence, recalling the definition of $M(x_k)$ and u^* and setting $\lambda_i^* = 0$ for $i \notin I_+(x^*)$, we obtain, (after reordering the components, if needed):

$$\nabla f(x^*) + \nabla g(x^*)\lambda^* + \nabla h(x^*)\mu^* = 0. \quad (21.11)$$

Moreover, as (λ_k, μ_k) converges to (λ^*, μ^*) and $\varepsilon_k \rightarrow 0$, recalling that

$$\lambda_k = 2/\varepsilon_k g^+(x_k), \quad \mu_k = 2/\varepsilon_k h(x_k),$$

we have $g^+(x_k) \rightarrow 0$ and $h(x_k) \rightarrow 0$, whence it follows that

$$g(x^*) \leq 0, \quad h(x^*) = 0,$$

which shows that x^* is a feasible point. As $\lambda^* \geq 0$ and $\lambda_i^* g_i(x^*) = 0$ for all i , we can conclude that the KKT conditions hold at x^* . \square

We note that in the preceding proposition we have not assumed *a priori* that the feasible set is non empty and hence the assumptions stated actually imply also feasibility of the constrained problem. Let us consider these assumptions in more detail.

As already discussed, assumption (i) is satisfied, for instance, if we use a globally convergent algorithm for unconstrained minimization and we suppose that Assumption 21.1 is valid.

Assumption (ii) requires that the sequence generated by the algorithm has a limit point in some compact set D . This is again an *a posteriori assumption* and, as already noted, it can be difficult, in general, to give *a priori conditions* that can guarantee its satisfaction. A quite restrictive condition, similar, in essence, to the conditions imposed on unconstrained methods, can be the following.

Assumption 21.2 *The feasible set S is non empty, a point $x_0 \in S$ is known and f is coercive on R^n .* \square

Under this assumption, we know that all level sets of the continuous function f are compact and, in particular, the level set

$$\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$$

is non-empty and compact. As

$$P(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \Psi(x)$$

we have also that $P(\cdot; \varepsilon)$ is coercive. Then, as Ψ is continuous and $\Psi(x_0) = 0$, it follows that the level set

$$\{x \in R^n : P(x; \varepsilon) \leq P(x_0; \varepsilon)\} = \{x \in R^n : f(x) + \frac{1}{\varepsilon}\Psi(x) \leq f(x_0)\}$$

is non empty and compact for every $\varepsilon > 0$. Thus, the instructions at Step 1 guarantee that $x_k \in \mathcal{L}_0$ for all k and we can assume $D = \mathcal{L}_0$.

Finally, Assumption (iii) first of all implies that the linear independence constraint qualification holds on the feasible set, but also excludes that the limit points that we obtain for $\varepsilon_k \rightarrow 0$, $\xi_k \rightarrow 0$ (which exist because of assumption (ii)) are *infeasible critical point of the penalty term*.

In the general case, if a feasible point is not available or the objective function is unbounded below, we should modify the merit function and introduce suitable additional conditions on the constraints. When this is impossible, in many cases we could attempt to remain in a bounded region by reducing the values of the penalty parameter ε .

Now we give some simple examples.

Example 21.1 Consider the problem of minimizing the function $f : R^2 \rightarrow R$ defined by $f(x) = x(1)^2 + 1/4x(2)^2$, under the constraint $x(1) = 2$. Obviously the unique solution is the point $x^* = (2, 0)$. The quadratic penalty function is the strictly convex quadratic function

$$P(x; \varepsilon) = x(1)^2 + 1/4x(2)^2 + \frac{1}{\varepsilon}(x(1) - 2)^2.$$

In this case we can compute exactly the unique stationary point and global minimizer of $P(x; \varepsilon_k)$, given by

$$x_k^* = \left(\frac{2}{1 + \varepsilon_k}, 0 \right).$$

This shows that for $\varepsilon_k \rightarrow 0$ the optimal solution will be reached in the limit.

Example 21.2 Consider the problem of minimizing the function $f : R \rightarrow R$, defined by $f(x) = x^3$, under the constraint $x = 1$. Obviously the unique minimizer is $x^* = 1$. A quadratic penalty function would be the function

$$P(x; \varepsilon) = x^3 + \frac{1}{\varepsilon}(x - 1)^2.$$

It is easily seen that, for any fixed ε , the function is unbounded below and hence does not admit an unconstrained minimizer. In practice, we can attempt, heuristically, to remain in a compact set, by employing sufficiently small values of the penalty parameter ε_k . \square

Example 21.3 Consider now the problem of minimizing on R the function $f(x) = x^2$, subject to $x^2 = -1$. The problem is obviously infeasible. The quadratic penalty function is

$$P(x; \varepsilon) = x^2 + \frac{1}{\varepsilon}(x^2 + 1)^2.$$

We can see that assumption (i) and (ii) are satisfied. In fact the only minimizer is $x_k = 0$ for every $\varepsilon > 0$, but assumption (iii) is not satisfied, because of the fact that the gradient of the constraint is $2x$ which is zero for $x = 0$. \square

We note, in general, that actually, assumption (iii) is somewhat stronger than that required for proving that a feasible point can be reached (*and hence that the feasible set is non empty*). In fact, if we minimize the penalty term, this can be established if we require that for some subsequence converging to $\bar{x} \in D$, the limit

$$\sum_{i=1}^m g_i^+(x_k) \nabla g_i(x_k) + \sum_{i=1}^p h_i(x_k) \nabla h_i(x_k) \rightarrow 0$$

implies $g_i^+(x_k) \rightarrow 0, i = 1, \dots, m$ and $h_i(x_k) \rightarrow 0, i = 1, \dots, p$.

This kind of condition will be further considered in connection with exact continuously differentiable penalty functions and Lagrangian functions.

21.3 Augmented Lagrangian Functions

The main problem in the use of sequential penalty functions is the need of employing in many cases large values of the penalty coefficient $1/\varepsilon$, which can produce ill-conditioning of the Hessian matrix of P . This typically has adverse effects on the convergence and on the convergence rate of algorithms for minimizing P , both when an unconstrained algorithm is directly employed and when P is used as a merit function for globalizing a local constrained algorithm. In order to overcome, to some extent, this difficulty, in the equality constrained case it has been proposed by Hestenes and Powell a new merit function that should guarantee a correspondence with the constrained problems for moderate values of $1/\varepsilon$. This function can be obtained either by adding a quadratic penalty term to the Lagrangian function, thus forming the so called *Augmented Lagrangian function*, or, equivalently, by shifting the penalty terms.

Consider the equality constrained problem

$$\min f(x), \quad h(x) = 0, x \in R^n \tag{21.12}$$

where $h : R^n \rightarrow R^p$.

The augmented Lagrangian of Hestenes for this problem is given by

$$L_a(x, \mu; \varepsilon) = f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2, \quad (21.13)$$

where $\mu \in R^p$ is an estimate of the Lagrangian multiplier.

The penalty function proposed by Powell can be put into the form

$$S(x, \mu; \varepsilon) = f(x) + \frac{1}{\varepsilon} \left\| h(x) + \frac{\varepsilon \mu}{2} \right\|^2. \quad (21.14)$$

It is easily seen that we have

$$S(x, \mu; \varepsilon) = L_a(x, \mu; \varepsilon) + \frac{\varepsilon}{4} \|\mu\|^2.$$

and hence the two functions are essentially equivalent, for fixed values of ε and μ . In the sequel we will refer to the augmented Lagrangian (21.13).

In the next proposition, making use of the second order sufficient conditions established in Chap. 5 we show that, under appropriate assumptions, a local minimizer of problem (21.12) is also a local minimizer of the augmented Lagrangian L_a , for sufficiently small positive values of ε , provided that μ is chosen as the Lagrange multiplier.

This motivates, essentially, the introduction of L_a .

Proposition 21.3 (Local Minimizers of the Augmented Lagrangian) *Let $x^* \in R^n$ be a local minimum point of problem (21.12) and suppose that the functions f, g, h are twice continuously differentiable on an open neighborhood of x^* .*

Suppose that (x^, μ^*) , with $\mu^* \in R^p$ satisfy the Lagrange multiplier rule and that the second order sufficient conditions of Proposition 5.6 hold.*

Then, there exists $\varepsilon^ > 0$ such that, for all $0 < \varepsilon \leq \varepsilon^*$ the point x^* is a local strict minimizer of $L_a(x, \mu^*; \varepsilon)$.*

Proof As the Lagrange multiplier rule holds at (x^*, μ^*) we have $\nabla_x L(x^*, \mu^*) = 0$ and $h(x^*) = 0$, so that

$$\nabla_x L_a(x^*, \mu^*; \varepsilon) = \nabla_x L(x^*, \mu^*) + \frac{2}{\varepsilon} \nabla h(x^*) h(x^*) = 0,$$

and hence x^* is a stationary point of $L_a(x, \mu^*; \varepsilon)$. We want to show that the Hessian matrix

$$\begin{aligned}\nabla_x^2 L_a(x^*, \mu^*; \varepsilon) &= \nabla_x^2 L(x^*, \mu^*) + \frac{2}{\varepsilon} \sum_{i=1}^p \left(h_i(x^*) \nabla^2 h_i(x^*) + \nabla h_i(x^*) \nabla h_i(x^*)^T \right) \\ &= \nabla_x^2 L(x^*, \mu^*) + \frac{2}{\varepsilon} \sum_{i=1}^p \nabla h_i(x^*) \nabla h_i(x^*)^T\end{aligned}\tag{21.15}$$

is positive definite for sufficiently small values of ε .

Reasoning by contradiction, if this assertion is false we can construct a sequence, for $k = 1, \dots$ such that $\varepsilon_k = 1/k$, $\|d_k\| = 1$ and

$$d_k^T \nabla_x^2 L_a(x^*, \mu^*; \varepsilon_k) d_k = d_k^T \nabla_x^2 L(x^*, \mu^*) d_k + \frac{2}{\varepsilon_k} \sum_{i=1}^p (\nabla h_i(x^*)^T d_k)^2 \leq 0,\tag{21.16}$$

which implies, in particular

$$d_k^T \nabla_x^2 L(x^*, \mu^*) d_k \leq 0 \quad \text{for all } k.\tag{21.17}$$

Now we can extract a subsequence, which we relabel $\{d_k\}$, such that d_k converges to some \bar{d} with $\|\bar{d}\| = 1$. Therefore, taking limits for $k \rightarrow \infty$, from (21.16) we get $\nabla h_i(x^*)^T \bar{d} = 0$ and from (21.17) we obtain $\bar{d}^T \nabla_x^2 L(x^*, \mu^*) \bar{d} \leq 0$ and this contradicts the assumption that the second order sufficient conditions are satisfied. \square

When f is convex and the functions h_i are affine, it easily seen that $L_a(x, \mu; \varepsilon)$ is convex, for every given $\mu \in R^p$ and $\varepsilon > 0$. Thus, if x^* is a global minimizer of Problem (21.12) and we assume that (x^*, μ^*) satisfies the Lagrange multiplier rule, we have that $\nabla_x L_a(x^*, \mu^*; \varepsilon) = 0$ and hence that x^* is a global minimizer of $L_a(x, \mu^*; \varepsilon)$ for every given $\varepsilon > 0$.

In the preceding results we have assumed that the augmented Lagrangian is evaluated in correspondence to the Lagrangian multiplier μ^* . In this case, for sufficiently large, but finite values of the penalty coefficient $1/\varepsilon$, we have seen that a minimizer of the constrained problem can be well approximated by a minimizer of L_a . This suggests the definition of an algorithm, called *multiplier method*, where, starting from some given initial estimate, the vector μ is updated at each step, in the attempt of approximating μ^* . If we consider the expression of ∇L_a , we can write

$$\nabla_x L_a(x, \mu; \varepsilon) = \nabla f(x) + \nabla h(x) \left(\mu + \frac{2}{\varepsilon} h(x) \right).$$

Therefore, as we are looking for a stationary point of $L(x, \mu)$ by minimizing L_a , the above expression suggests an updating rule for μ of the form

$$\mu_{k+1} = \mu_k + \frac{2}{\varepsilon_k} h(x_k),$$

where μ_k, x_k are the current estimates. In this case a sequence of unconstrained problems can be solved by minimizing approximately the augmented Lagrangian $L_a(x, \mu_k, \varepsilon_k)$ for each k , thus obtaining the point x_k , and then updating the penalty coefficient ε_k and the Lagrangian multiplier estimate μ_k . Under appropriate assumptions, we can define a technique such that convergence to critical points of the constrained problem can be guaranteed without the need of imposing that $\varepsilon_k \rightarrow 0$ or, at least, with an improved convergence speed in comparison with the sequential penalty function method.

The extension of this technique to the case of inequality constraints has been defined, for the first time, by Rockafellar and can be performed by transforming each inequality constraint $g_i(x) \leq 0$ into an equality constraint, through a squared slack variable. Thus, with reference to problem (21.5), we obtain the function:

$$\begin{aligned} \hat{L}_a(x, y, \lambda, \mu; \varepsilon) &= f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2 \\ &\quad + \sum_{i=1}^m \lambda_i (g_i(x) + y_i^2) + \frac{1}{\varepsilon} \sum_{i=1}^m (g_i(x) + y_i^2)^2, \end{aligned} \tag{21.18}$$

where $y \in R^m$ is the vector with components $y_i, i = 1, \dots, m$.

The function should be minimized with respect to x, y for fixed values of λ, μ and the minimization with respect to y can be performed analytically. Actually, as the function is separable with respect to the components y_i , we can minimize each scalar term

$$\phi_i(y_i) \equiv \lambda_i (g_i(x) + y_i^2) + \frac{1}{\varepsilon} (g_i(x) + y_i^2)^2.$$

This function is clearly continuously differentiable with respect to y_i ; moreover, as $\phi_i(y_i) \rightarrow \infty$ for $|y_i| \rightarrow \infty$, it admits a global minimizer in R .

By imposing the first order optimality conditions we have

$$y_i \left(\lambda_i + \frac{2}{\varepsilon} g_i(x) + \frac{2}{\varepsilon} y_i^2 \right) = 0.$$

It is easily seen that a stationary point is $\bar{y}_i = 0$ and, in this case, we have

$$\phi_i(\bar{y}_i) = \lambda_i g_i(x) + \frac{1}{\varepsilon} g_i(x)^2 \geq -\frac{\varepsilon \lambda_i^2}{4}. \tag{21.19}$$

We note that the last inequality can be obtained by taking the minimum value of $\phi_i(\bar{y}_i)$ with respect to all possible values of $g_i(x)$. From the optimality conditions we have that another stationary point can be obtained if there exists \hat{y}_i such that

$$\hat{y}_i^2 = -\left(\frac{\varepsilon\lambda_i}{2} + g_i(x)\right),$$

which is possible if and only if $\varepsilon\lambda_i/2 + g_i(x) \leq 0$. In this case we have $g_i(x) + \hat{y}_i^2 = -\varepsilon\lambda_i/2$ and hence we obtain

$$\phi_i(\hat{y}_i) = -\frac{\varepsilon\lambda_i^2}{4},$$

so that, recalling (21.19) we have that \hat{y}_i is a global minimizer.

Then we can assume as optimal solution the value $y_i(x, \lambda_i, \varepsilon)$ such that

$$y_i^2(x, \lambda_i, \varepsilon) = \max \left[0, -\left(\frac{\varepsilon\lambda_i}{2} + g_i(x)\right) \right],$$

which implies

$$g_i(x) + y_i^2(x, \lambda_i, \varepsilon) = \max [g_i(x), -\varepsilon\lambda_i/2]. \quad (21.20)$$

Recalling (21.18), the merit function becomes:

$$\begin{aligned} L_a(x, \lambda, \mu; \varepsilon) &= f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2 \\ &\quad + \sum_{i=1}^m \lambda_i \left(g_i(x) + y_i^2(x, \lambda_i, \varepsilon) \right) + \frac{1}{\varepsilon} \sum_{i=1}^m \left(g_i(x) + y_i^2(x, \lambda_i, \varepsilon) \right)^2, \end{aligned} \quad (21.21)$$

It can be easily verified that we can put the augmented Lagrangian in the form

$$\begin{aligned} L_a(x, \lambda, \mu; \varepsilon) &= f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2 \\ &\quad + \frac{1}{\varepsilon} \sum_{i=1}^m \left[\max \left[0, g_i + \frac{\varepsilon\lambda_i}{2} \right] \right]^2 - \frac{\varepsilon}{4} \sum_{i=1}^m \lambda_i^2. \end{aligned} \quad (21.22)$$

We note that the function is continuously differentiable with respect to x and we can write the following expression for the gradient $\nabla_x L_a$.

$$\begin{aligned}\nabla_x L_a(x, \lambda, \mu; \varepsilon) &= \nabla f(x) + \nabla h(x)\mu + \frac{2}{\varepsilon} \nabla h(x)h(x) \\ &\quad + \frac{2}{\varepsilon} \sum_{i=1}^m \max \left[0, g_i + \frac{\varepsilon \lambda_i}{2} \right] \nabla g_i(x).\end{aligned}\tag{21.23}$$

This function can also be rewritten into the form

$$\begin{aligned}L_a(x, \lambda, \mu; \varepsilon) &= f(x) + \frac{1}{\varepsilon} \sum_{i=1}^p \left[h_i + \frac{\varepsilon \mu_i}{2} \right]^2 \\ &\quad + \frac{1}{\varepsilon} \sum_{i=1}^m \left[\left(g_i + \frac{\varepsilon \lambda_i}{2} \right)^+ \right]^2 - \frac{\varepsilon}{4} \sum_{i=1}^p \mu_i^2 - \frac{\varepsilon}{4} \sum_{i=1}^m \lambda_i^2,\end{aligned}\tag{21.24}$$

which shows the penalty shifts in the augmented Lagrangian.

In fact, the extension of the shifted penalty function of Powell to the problem with equality and inequality constraints is given by

$$S(x, \lambda, \mu; \varepsilon) = f(x) + \frac{1}{\varepsilon} \sum_{i=1}^p \left[h_i + \frac{\varepsilon \mu_i}{2} \right]^2 + \frac{1}{\varepsilon} \sum_{i=1}^m \left[\left(g_i + \frac{\varepsilon \lambda_i}{2} \right)^+ \right]^2.\tag{21.25}$$

The updating rule for the multipliers can be defined according to the following rules.

$$\mu_{k+1} = \mu_k + \frac{2}{\varepsilon_k} h(x_k),$$

$$\lambda_{k+1} = \left(\lambda_k + \frac{2}{\varepsilon_k} g(x_k) \right)^+.$$

On the basis of the preceding formulae we can relate stationary points of L_a with KKT points of the constrained problem. We state this relationship as a proposition.

Proposition 21.4 (Relationships Between Critical Points) Suppose that all problem functions are continuously differentiable and let $\bar{x} \in R^n$, $\bar{\lambda} \in R^m$, $\bar{\mu} \in R^p$ be such that $\nabla L_a(\bar{x}, \bar{\lambda}, \bar{\mu}; \varepsilon) = 0$, $h(\bar{x}) = 0$, $\bar{\lambda} \geq 0$ and

$$g_i(\bar{x}) + y_i^2(\bar{x}, \bar{\lambda}; \varepsilon) = 0 \quad i = 1, \dots, m.$$

Then, $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is a KKT triple for the constrained problem.

Proof The assumptions stated obviously imply that \bar{x} is feasible. If

$$g_i(\bar{x}) + y_i^2(\bar{x}, \bar{\lambda}; \varepsilon) = 0$$

for all i we have, by (21.20) that

$$\max [g_i(\bar{x}), -\varepsilon \bar{\lambda}_i / 2] = 0$$

for all i . This implies that we must have necessarily that $\bar{\lambda}_i g_i(\bar{x}) = 0$ for all i . Finally, taking this into account and recalling (21.23) we have that $\nabla L(\bar{x}, \bar{\lambda}, \bar{\mu}) = 0$.

□

On the basis of the preceding results we can define a simplified conceptual model of the multiplier method, partially based on [23] and on Proposition 21.2, where we impose that the multipliers remain bounded and that the penalty parameter is updated according to a suitable rule.

In this scheme we define a criterion for testing feasibility and complementarity by evaluating for each k the number

$$T_k = \|h(x_k)\|_\infty + \max_{i=1,m} |g_i(x_k) + y_i^2(x_k, \lambda_k; \varepsilon_k)|. \quad (21.26)$$

By Proposition 21.4, we know that $T_k = 0$ and

$$\nabla L_a(x_k, \lambda_k, \mu_k; \varepsilon_k) = 0$$

imply that we have reached a KKT point.

Algorithm 21.2 (Multiplier Method)

Data $\varepsilon_1 > 0$, $\tau \in (0, 1)$, $\theta \in (0, 1)$, $\mu_{\min} < \mu_{\max}$, $\lambda_{\max} > 0$, $T_0 = 0$.
Choose $x_0 \in R^n$, $\mu_1 \in [\mu_{\min}, \mu_{\max}]^p$ and $\lambda_1 \in [0, \lambda_{\max}]^m$ and define a sequence $\{\xi_k\}$ such that $\xi_k > 0$ and $\lim_{k \rightarrow \infty} \xi_k = 0$. Set $k = 1$

For k=1, 2, ...

1. Using an unconstrained minimization algorithm, starting from x_{k-1} (or x_0) determine x_k such that $\|\nabla L_a(x_k, \lambda_k, \mu_k; \varepsilon_k)\| \leq \xi_k$.
2. Compute T_k ; if $T_k \leq \tau T_{k-1}$ set $\varepsilon_{k+1} = \varepsilon_k$; else set $\varepsilon_{k+1} = \theta \varepsilon_k$.
3. Perform a safeguarded update of the multipliers. Compute the tentative vectors $\tilde{\mu}_{k+1} = \mu_k + 2/\varepsilon_k h(x_k)$, $\tilde{\lambda}_{k+1} = (\lambda_k + 2/\varepsilon_k g(x_k))^+$ and modify, when needed, the components in a way that $\mu_{k+1} \in [\mu_{\min}, \mu_{\max}]^p$ and $\lambda_{k+1} \in [0, \lambda_{\max}]^m$

End for

As the preceding scheme guarantees that the multipliers remain bounded, by construction, we can analyze (and, possibly, modify) the algorithm along similar lines to those followed in the case of Algorithm 21.1 and we can extend the convergence proof of Proposition 21.2.

21.4 Non Differentiable Exact Penalty Functions

The essential feature of this class of functions is the definition of a non differentiable penalty term on the constraint violation, typically represented by an ℓ -norm. The best known class of these functions can be defined by:

$$J_q(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \left(\sum_{i=1}^m (g_i^+(x))^q + \sum_{i=1}^p |h_i(x)|^q \right)^{1/q}, \quad (21.27)$$

for $1 \leq q < \infty$ and

$$J_\infty(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \max [g_1^+(x), \dots, g_m^+(x), |h_1(x)|, \dots, |h_p(x)|].$$

The function J_q is not differentiable at feasible points and the presence of a non differentiable penalty term in a penalty function based on an ℓ -norm is, in general, a necessary condition to keep bounded the penalty coefficient $1/\varepsilon$. In fact, if we consider a differentiable penalty function, such as the quadratic penalty function, we have that the gradient of the penalty term is zero at feasible points and hence it is quite exceptional the possibility of reaching a KKT point for a fixed $\varepsilon > 0$, as we should have also $\nabla f = 0$.

Under suitable assumptions, we can establish a correspondence between critical points and minimizers of J_q and KKT points and minimizers of the constrained problems for finite values of the penalty coefficient and we refer to the literature for basic theoretical results.

The computational use of these functions will be discussed in the next chapter, in connection with recursive quadratic programming algorithms.

21.5 Continuously Differentiable Exact Penalty Functions

Continuously differentiable exact penalty functions consists in replacing the multiplier vectors λ, μ which appear in the augmented Lagrangian function of Hestenes, Powell and Rockafellar with continuously differentiable *multiplier functions* such that $\lambda : R^n \rightarrow R^m, \mu : R^n \rightarrow R^p$, and that, if $\bar{x}, \bar{\lambda}, \bar{\mu}$ satisfy the KKT conditions, we have $\lambda(\bar{x}) = \bar{\lambda}, \mu(\bar{x}) = \bar{\mu}$.

In the equality constrained case, assuming that the gradients $\nabla h_i(x)$, $i = 1, \dots, p$ are linearly independent at every $x \in R^n$, the multiplier function $\mu(\cdot)$ can be obtained by minimizing with respect to μ the quadratic function

$$\psi(\mu; x) = \|\nabla f(x) + \nabla h(x)\mu\|^2.$$

This yields:

$$\mu(x) = -\left(\nabla h(x)^T \nabla h(x)\right)^{-1} \nabla h(x)^T \nabla f(x).$$

By substituting this expression into the augmented Lagrangian, we get the exact penalty function proposed by Fletcher.

$$W(x; \varepsilon) = f(x) + \mu(x)^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2. \quad (21.28)$$

Under suitable assumptions, it can be shown that there exists a threshold value of the parameter, ε^* , such that for $0 < \varepsilon \leq \varepsilon^*$, we can establish, on a compact region, a correspondence between stationary points and minimizers of W and critical points and minimizers of the constrained problems.

Example 21.4 Consider again Example 21.1, that is the problem of minimizing the function $f : R^2 \rightarrow R$ defined by $f(x) = x(1)^2 + 1/4x(2)^2$, under the constraint $h(x) \equiv x(1) - 2 = 0$. If we impose the condition $\nabla L(x, \mu) = 0$ we get the multiplier function $\mu(x) = -2x(1)$. Therefore, the exact penalty function becomes

$$W(x; \varepsilon) = x(1)^2 + 1/4x(2)^2 - 2x(1)(x(1) - 2) + \frac{1}{\varepsilon}(x(1) - 2)^2.$$

It is easily seen that for $\varepsilon \leq \varepsilon^* < 1$, say $\varepsilon = 1/2$, the optimal solution is $(2, 0)$ and this proves exactness. \square

The extension to problems with inequality constraints can be obtained by defining appropriately the multiplier functions and by employing squared slack variables for transforming inequality constraints into equalities, as we did in the case of the augmented Lagrangian.

To simplify notation, let us assume, without loss of generality, that the problem has only inequality constraints, so that we have:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \end{aligned} \quad (21.29)$$

where $f : R^n \rightarrow R$ and $g : R^n \rightarrow R^m$ are twice continuously differentiable.

First we define the multiplier function, as proposed in [116], by minimizing with respect to λ the quadratic function defined by:

$$\Phi(\lambda; x) = \|\nabla_x L(x, \lambda)\|^2 + \gamma^2 \|G(x)\lambda\|^2,$$

where $\gamma > 0$ and $G(x) = \text{Diag}(g(x))$. Assuming that the gradients of active constraints at x are linearly independent, the Hessian matrix of Φ , given by:

$$N(x) = \nabla g(x)^T \nabla g(x) + \gamma^2 G^2(x),$$

is positive definite and hence the multiplier function can be represented in the form

$$\lambda(x) = -N^{-1}(x) \nabla g(x)^T \nabla f(x).$$

It can be easily verified that, if $(\bar{x}, \bar{\lambda})$ is a KKT pair, then $\lambda(\bar{x})$ coincides with the KKT multiplier vector. In fact, we have

$$\Phi(\bar{\lambda}; \bar{x}) = 0,$$

which obviously gives the minimum value of $\Phi \geq 0$ and hence, as we have a unique minimizer, we must have necessarily

$$\lambda(\bar{x}) = \bar{\lambda}.$$

Inequality constraints can be transformed into equality constraints by employing squared slack variables and minimizing analytically the penalty function with respect to the slack variables. This yields the continuously differentiable function:

$$\begin{aligned} U(x; \varepsilon) &= f(x) + \lambda(x)^T (g(x) + Y(x : \varepsilon)y(x : \varepsilon)) \\ &\quad + \frac{1}{\varepsilon} \|g(x) + Y(x : \varepsilon)y(x : \varepsilon)\|^2, \end{aligned} \tag{21.30}$$

where

$$\begin{aligned} y_i(x : \varepsilon) &= \{-\min[0, g_i(x) + (\varepsilon/2)\lambda_i(x)]\}^{1/2}, \quad i = 1, \dots, m \\ Y(x : \varepsilon) &= \text{Diag}(y(x : \varepsilon)). \end{aligned} \tag{21.31}$$

It can be shown [72] that the function is continuously differentiable and that we can establish exactness properties, for sufficiently small values of the penalty parameter ε on some given compact set $D \subset R^n$, provided that the solutions of interest are in the interior of D and that suitable regularity conditions on the constraints are satisfied.

21.6 Exact Shifted Barrier Functions

As already discussed, the compactness requirement on the set where the unconstrained minimization is carried out is a common feature of all penalty methods, and may cause, in principle, computational difficulties.

In particular, in the case of exact penalty functions, the level set of the penalty function corresponding to some initial point and some value of the penalty parameter, even if compact, may be not contained in the set D where exactness is established. Therefore a minimizing sequence starting from a given point in D could be attracted towards stationary points of the penalty function outside the set D or may not converge at all if the level set of the penalty function is unbounded. An example of this situation is given below.

Example 21.5 Consider again problem 21.2, that is the problem of minimizing the function $f : R \rightarrow R$, defined by $f(x) = x^3$, under the constraint $x = 1$.

A continuously differentiable exact penalty function can be constructed by employing the multiplier function $\mu(x) = -3x^2$, (which follows immediately from $\nabla_x L(x, \mu) = 0$), so that we get

$$W(x; \varepsilon) = x^3 - 3x^2(x - 1) + \frac{1}{\varepsilon}(x - 1)^2.$$

Letting $\nabla W(x; \varepsilon) = 0$ we can impose, after simple manipulations, that

$$\nabla W(x; \varepsilon) = (1 - x)(3x - 1/\varepsilon) = 0.$$

This implies that, for every $\varepsilon > 0$, the function has the stationary point $x^* = 1$, which is the optimal solution of the constrained problem, but also the *spurious* stationary point $\hat{x} = 1/(3\varepsilon)$. The reason is that the function W has unbounded level set on R , but correspondence of critical points and minimizers for sufficiently small values of $\varepsilon > 0$ can be established only on some compact set. In practice, we can choose a small value of ε , and attempt to remain in a compact set during the unconstrained minimization.

In Fig. 21.1 we show the behaviour of the function W for two values of ε , namely for $\varepsilon = 1$ and $\varepsilon = 0.1$.

We can note that for $\varepsilon = 1$ the point $x^* = 1$ is a stationary point, but it is a local maximizer, while $x = 1/3$ is a local minimizer.

For $\varepsilon = 0.1$ the spurious stationary point is moved to $x = 10/3$ and the point $x^* = 1$ becomes a local minimizer. The function would be an exact penalty function for $\varepsilon = 0.1$ only if we restrict our search to an interval with right endpoint smaller than $10/3$. \square

Under suitable assumptions on the problem, it is possible to avoid (or mitigate) the difficulty discussed above, by introducing suitable *shifted barrier terms* on the constraints. These techniques differ from the interior point methods described in

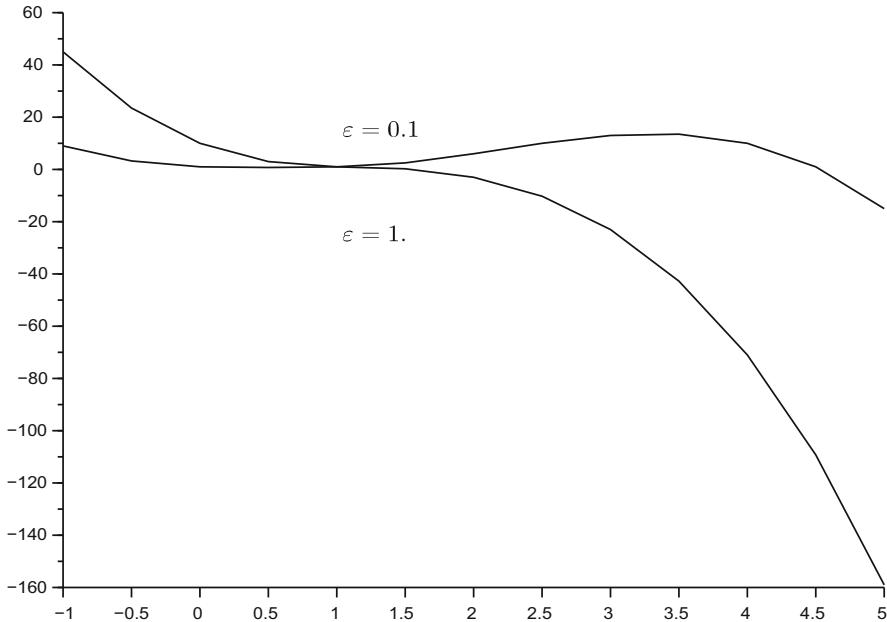


Fig. 21.1 Function W for $\varepsilon = 1$ and $\varepsilon = 0.1$

Chap. 23. In fact, thanks to the shifts, we can reach exactly the boundary of the feasible set using *exterior methods*.

Here we will consider only the exact penalty functions based on shifted barrier techniques, which have been introduced in [73]. These results have been improved and specialized in various works. In particular, the short introduction given here is based essentially on [170], with some (minor) modifications introduced for simplifying our exposition.

We refer again to a continuously differentiable penalty function for solving the problem with only inequality constraints

$$\begin{aligned} \min f(x) \\ g(x) \leq 0, \end{aligned} \tag{21.32}$$

where $f : R^n \rightarrow R$ and $g : R^n \rightarrow R^m$ are assumed to be twice continuously differentiable. We denote by $S = \{x \in R^n : g(x) \leq 0\}$ the feasible set and we define the index sets $I_0(x) = \{i : g_i(x) = 0\}$, and $I_+(x) = \{i : g_i(x) \geq 0\}$.

Now we consider a perturbation of the feasible set, by defining the open set

$$D_0 = \{x \in R^n : \sum_{i=1}^m g_i^+(x)^p < \alpha\},$$

where $p \geq 2$ and $\alpha > 0$. It is easily seen that the feasible set is contained in D_0 . We can observe that the parameter α can always be chosen so that D_0 is non empty. It is only required to fix a point \tilde{x} in R^n and then to assume $\alpha > 0$ in a way that

$$\alpha > \sum_{i=1}^m g_i^+(\tilde{x})^p.$$

We define the function

$$a(x) = \alpha - \sum_{i=1}^m g_i^+(x)^p$$

and we suppose that the following assumptions hold.

Assumption 21.3

A1. *The set $D = \text{Cl}(D_0)$ is compact.*

A2. *At every $x \in S$ the gradients $\nabla g_i(x)$, $i \in I_0(x)$ are linearly independent.*

□

Now consider the multiplier function $\lambda(x)$ introduced in [170], which is obtained by minimizing the quadratic function in λ :

$$\Phi(\lambda; x) = \|\nabla_x L(x, \lambda)\|^2 + \gamma_1 \|G(x)\lambda\|^2 + \gamma_2 s(x)\|\lambda\|^2,$$

where $\gamma_1, \gamma_2 > 0$, $G(x) = \text{Diag}(g(x))$ and $s(x) = \|g^+(x)\|^p$.

Under Assumption A2, it is easily seen that the Hessian matrix N of Φ , given by

$$N(x) = \nabla g(x)^T \nabla g(x) + \gamma_1 G^2(x) + \gamma_2 s(x) I_m,$$

is positive definite on D . Then the multiplier function is given by

$$\lambda(x) = -N^{-1}(x) \nabla g(x)^T \nabla f(x).$$

Now we can construct an augmented Lagrangian function, using the multiplier function defined above and replacing the penalty coefficient $1/\varepsilon$ with the shifted barrier term $1/(\varepsilon a(x))$.

Employing the squared slack variable approach as in the preceding section and minimizing with respect to the slack variables, we obtain the function

$$\begin{aligned} Z(x; \varepsilon) &= f(x) + \lambda(x)^T (g(x) + Y(x : \varepsilon)y(x : \varepsilon)) \\ &+ \frac{1}{\varepsilon a(x)} \|g(x) + Y(x : \varepsilon)y(x : \varepsilon)\|^2, \end{aligned} \tag{21.33}$$

where

$$y_i(x : \varepsilon) = \left\{ -\min \left[0, g_i(x) + \frac{\varepsilon a(x)}{2} \lambda_i(x) \right] \right\}^{1/2}, \quad i = 1, \dots, m \quad (21.34)$$

$$Y(x : \varepsilon) = \text{Diag}(y(x : \varepsilon)).$$

In order to simplify notation, we omit the dependence on ε and we set $Z \equiv Z(\cdot; \varepsilon)$. The function Z is defined on D_0 and it can be shown that the following proposition holds.

Proposition 21.5 (Basic Properties of Z) *Suppose that all problem functions are twice continuously differentiable on D_0 ; then, for every $\varepsilon > 0$, we have:*

- (i) Z is continuously differentiable on D_0 ;
- (ii) $Z(x; \varepsilon) \leq f(x)$ for all $x \in S$;
- (iii) Z admits a global (unconstrained) minimum point in D_0 .

Proof Points (i) and (ii) follow from the expression of Z . To establish (iii) we must prove that a level set of Z corresponding to some $\tilde{x} \in D_0$, that is

$$\mathcal{L}_Z = \{x \in D_0 : Z(x; \varepsilon) \leq Z(\tilde{x}; \varepsilon)\},$$

is compact. By assumption A1, the set D is compact and hence $\mathcal{L}_Z \subset D_0$ is bounded. We must prove that the level set is closed. Assume the contrary, then, there must exist some sequence of points $x_k \in \mathcal{L}_Z$ converging to a point \hat{x} at the boundary of D . This would imply that $a(x_k) \rightarrow a(\hat{x}) = 0$ and hence that

$$\|g^+(\hat{x})\|^p = \alpha > 0.$$

On the other hand, recalling that x_k remains in the bounded level set and that all problem functions are continuous, it is easily seen that $a(x_k) \rightarrow 0$ implies that $a(x_k)Z(x_k : \varepsilon) \rightarrow 0$ and this in turn implies that $\|g^+(\hat{x})\| = 0$, which yields a contradiction. \square

In order to illustrate the construction of Z , let us consider a simple example, which can be viewed as an inequality version of the problem of Example 21.5.

Example 21.6 Consider the problem of minimizing the function $f : R \rightarrow R$, defined by $f(x) = x^3$, under the box constraint

$$g_1(x) \equiv 1 - x \leq 0, \quad g_2(x) \equiv x - 2 \leq 0.$$

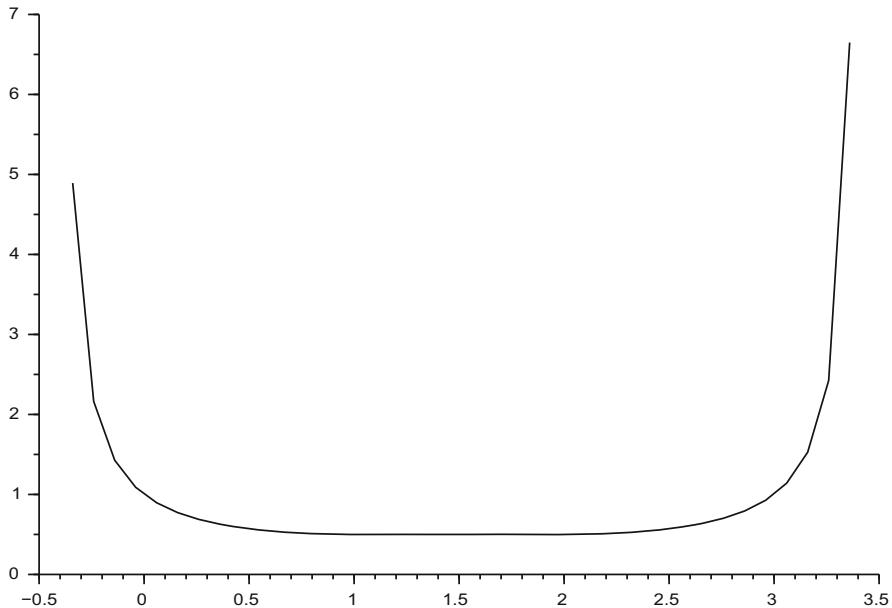


Fig. 21.2 Function $b(x) = 1/a(x)$

First of all, we construct the set $D_0 \subset R$. We fix a point in R , say $\bar{x} = 0$, and we assume $p = 2$. Thus we get

$$(g_1^+)^2(\bar{x}) + (g_2^+)^2(\bar{x}) = 1,$$

so that we can choose, for instance, $\alpha = 2$ and we obtain

$$D_0 = \{x \in R : (\max\{0, 1-x\})^2 + (\max\{0, x-2\})^2 < 2\},$$

which is obviously bounded. Then we obtain the function

$$a(x) = 2 - (\max\{0, 1-x\})^2 - (\max\{0, x-2\})^2,$$

which will be used for constructing the barrier. In Fig. 21.2 we show the behaviour of the barrier function $b(x) = 1/a(x)$.

It can be verified that the set D_0 is the open interval (l, u) with endpoints:

$$l = 1 - \sqrt{2}, \quad u = 2 + \sqrt{2}.$$

Because of the very simple structure of the constraints, on the basis of the KKT conditions, we can assume as multiplier function the function

$$\lambda(x) = \begin{pmatrix} 3x^2 \\ 0 \end{pmatrix}.$$

Then, we can set:

$$g_1 + y_1^2 = 1 - x - \min\{0, 1 - x + \varepsilon \frac{a(x)}{2} 3x^2\}$$

$$g_2 + y_2^2 = x - 2 - \min\{0, x - 2\}$$

and the exact penalty function becomes:

$$Z(x; \varepsilon) = x^3 + 3x^2(g_1 + y_1^2) + \frac{1}{\varepsilon a(x)} ((g_1 + y_1^2)^2 + (g_2 + y_2^2)^2).$$

The behaviour of f and Z , for $\varepsilon = 0.05$ is shown in Fig. 21.3.

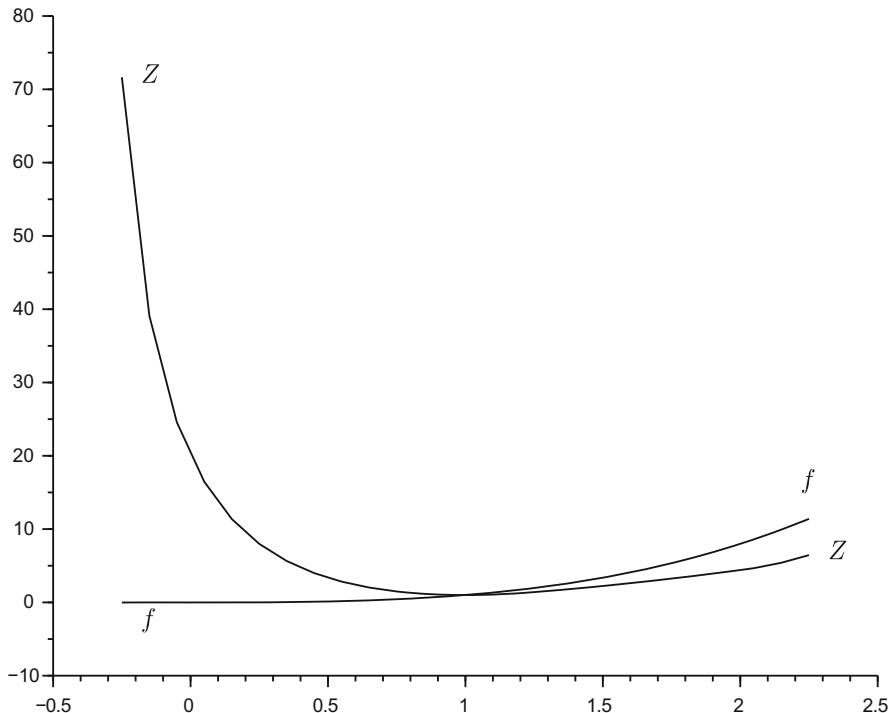


Fig. 21.3 Functions f and Z ($\varepsilon = 0.05$)

We can note that $f(x) \geq Z(x; \varepsilon)$ on the feasible set and that the optimal solution $x^* = 1$ of the constrained problem is the unique minimizer of Z . \square

In order to state the main exactness results we must introduce a further assumption, which essentially implies that the feasible set is non empty, and that will be explicitly invoked when required.

Assumption 21.4

A3. For $x \in D_0$, letting $\psi(x) = \frac{\|g(x)^+\|^p}{a(x)}$, with $p \geq 2$, we have that

$$\nabla\psi(x) = 0 \quad \text{implies that} \quad g(x)^+ = 0..$$

\square

It can be verified that assumption A3 is satisfied, provided that the *Extended Mangasarian-Fromovitz condition* (EMF) holds on D .

Extended Mangasarian-Fromovitz Condition

For all $x \in D$ we have that

$$\sum_{i \in I_+(x)} u_i \nabla g_i(x) = 0, \quad u_i \geq 0, \quad i \in I_+(x), \quad \text{implies that } u_i = 0, \quad i \in I_+(x).$$

\square

We have

$$\nabla\psi(x) = \frac{2}{a(x)} \sum_{i=1}^m \left[1 + \frac{p}{2} \frac{\|g(x)^+\|^{p-2}}{a(x)} \right] g_i^+(x) \nabla g_i(x)$$

and hence it is easily seen that if the EMF holds assumption A3 is satisfied.

Now we can summarize the main properties of the exact penalty function Z . The proofs omitted here because of space limitations and additional results can be found in [170]. The next proposition states some simple relationships between critical points.

Proposition 21.6 *For every $\varepsilon > 0$, we have:*

- (i) *if $(\bar{x}, \bar{\lambda})$ is a KKT pair for the constrained problem then*

$$g(x) + Y(x : \varepsilon)y(x : \varepsilon) = 0, \quad Z(\bar{x}; \varepsilon) = f(\bar{x}) \text{ and } \nabla Z(\bar{x}; \varepsilon) = 0;$$
- (ii) *if $\bar{x} \in D_0$ satisfies $\nabla Z(\bar{x}; \varepsilon) = 0$ and $g(\bar{x}) + Y(\bar{x} : \varepsilon)y(\bar{x} : \varepsilon) = 0$, then*

$$(\bar{x}, \lambda(\bar{x})) \text{ is a KKT pair for the constrained problem.} \quad \square$$

The key result for establishing exactness is the following proposition. We must assume either that a feasible point is available, and hence we can start the unconstrained minimization from this point, or that Assumption A3 is satisfied and, in this case, we can start our search from any point in D_0 .

Proposition 21.7 (Relationships Between Critical Points) *Let $\tilde{x} \in D_0$ and suppose that $\tilde{x} \in S$ or that Assumption A3 holds on D . Then there exists $\varepsilon^* > 0$ such that, for all $0 < \varepsilon \leq \varepsilon^*$, if the point*

$$\hat{x} \in \mathcal{L}_Z = \{x \in D_0 : Z(x; \varepsilon) \leq Z(\tilde{x}; \varepsilon)\}$$

is a stationary point of $Z(\cdot; \varepsilon)$, the pair $(\hat{x}, \lambda(\hat{x}))$ is a KKT pair for the constrained problem. \square

The next propositions give the main optimality results. First of all we can guarantee that we do not miss any global minimizer of the constrained problem.

Proposition 21.8 (Correspondence of Global Minimizers) *Let $x^* \in S$ be a global minimizer of the constrained problem. Then there exists $\varepsilon^* > 0$ such that, for all $0 < \varepsilon \leq \varepsilon^*$, the point x^* is a global minimum point of $Z(\cdot; \varepsilon)$ on D_0 .*

Proof Suppose that $x^* \in S$ is a global minimum point of the constrained problem. By assumption A2 the linear independence constraint qualification is satisfied and hence there exists λ^* such that the pair (x^*, λ^*) satisfies the KKT conditions.

Then it follows from Proposition 21.6 that $Z(x^*; \varepsilon) = f(x^*)$ for every $\varepsilon > 0$. Suppose now that $\varepsilon \in (0, \varepsilon^*]$, where ε^* is the number considered in Proposition 21.7. The assumptions of this proposition are obviously satisfied if we choose x^* as \tilde{x} .

Now suppose that our assertion is false. Then there must exist $\hat{x} \in D_0$ which is a global minimizer of Z such that

$$Z(\hat{x}; \varepsilon) < Z(x^*; \varepsilon) = f(x^*) \quad \text{and that} \quad \nabla Z(\hat{x}; \varepsilon) = 0.$$

However, by Proposition 21.7 the pair $(\hat{x}, \lambda(\hat{x}))$ must be a KKT pair for the constrained problem and by Proposition 21.6 we have that $Z(\hat{x}; \varepsilon) = f(\hat{x})$, but hence we should have $f(\hat{x}) < f(x^*)$, which contradicts the assumption that x^* was a global minimizer. \square

The converse result guarantees that we do not introduce spurious global or local minimizers of Z .

Proposition 21.9 (Correspondence of Minimizers of Z) *Let $\tilde{x} \in D_0$ and suppose that $\tilde{x} \in S$ or that Assumption A3 holds on D . Then there exists $\varepsilon^* > 0$ such that, for all $0 < \varepsilon \leq \varepsilon^*$, if the point*

$$x^* \in \mathcal{L}_Z = \{x \in D_0 : Z(x; \varepsilon) \leq Z(\tilde{x}; \varepsilon)\}$$

is a local (global) unconstrained minimum point of $Z(\cdot; \varepsilon)$, then x^ is a local (global) minimum point for the constrained problem and $\lambda(x^*)$ is the associated KKT multiplier.*

Proof Suppose first that $x^* \in D_0$ is a local unconstrained minimizer (and hence a stationary point) of $Z(\cdot; \varepsilon)$ for $\varepsilon \in (0, \varepsilon^*]$, where ε^* is the number considered in Proposition 21.7. By Proposition 21.7 the pair $(x^*, \lambda(x^*))$ is a KKT pair for the constrained problem. Then, by Proposition 21.6 we have that $Z(x^*; \varepsilon) = f(x^*)$. As x^* is a local minimizer of $Z(\cdot; \varepsilon)$ there exists a neighborhood $\mathcal{Q} \subset D_0$ of x^* , such that

$$f(x^*) = Z(x^*; \varepsilon) \leq Z(x; \varepsilon) \text{ for all } x \in \mathcal{Q}.$$

Therefore, recalling (ii) of Proposition 21.5 we have also $Z(x; \varepsilon) \leq f(x)$ for all $x \in S$ and hence we have

$$f(x^*) = Z(x^*; \varepsilon) \leq Z(x; \varepsilon) \leq f(x) \text{ for all } x \in \mathcal{Q} \cap S,$$

which establishes our thesis.

Suppose now that $x^* \in D_0$ is a global unconstrained minimizer of $Z(\cdot; \varepsilon)$ for $\varepsilon \in (0, \varepsilon^*]$. We can repeat the same reasoning, replacing \mathcal{Q} with the whole D_0 . \square

21.7 Exact Augmented Lagrangian Functions

In the general, unstructured case, the main disadvantage of continuously differentiable exact penalty functions is the need of solving a $m \times m$ linear system for computing the multiplier function each time that the penalty function must be evaluated. When m is very large this could be quite expensive from a computational point of view. An alternative approach to the definition of exact penalty methods is that of introducing an augmented Lagrangian such that, under appropriate assumptions, for sufficiently small values of the penalty parameter $\varepsilon > 0$, the unconstrained minimization of this function with respect to both the problem variables and the multiplier estimates yields solutions of the constrained problem.

To illustrate this possibility, let us first consider the equality constrained problem

$$\begin{aligned} \min_{x} f(x) \\ h(x) = 0, \end{aligned}$$

where $h : R^n \rightarrow R^p$. In this case we could construct a merit function depending on the pair x, μ , by penalizing the first order optimality conditions. In particular, we could consider the problem:

$$\min_{x, \mu} \| \nabla_x L(x, \mu) \|^2 + \| h(x) \|^2. \quad (21.35)$$

When h is affine, f is convex and we can find a global solution x^*, μ^* of the unconstrained problem, we have that the optimal objective function value in (21.35) is zero if and only if the pair x^*, μ^* satisfies the Lagrange multiplier rule. In this case, because of convexity, x^* is a global solution of the constrained problem.

In the general, non convex case, the penalization of the necessary optimality conditions does not induce any preference towards minimizers rather than maximizers of the original objective function. However, we can construct an augmented Lagrangian function $A(x, \mu; \varepsilon)$ that can ensure, under appropriate assumptions, a correspondence of unconstrained minimizers of $A(x, \mu; \varepsilon)$ with constrained minimizers and associated Lagrange multipliers.

In particular, we can consider the function:

$$A(x, \mu; \varepsilon) = f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \| h(x) \|^2 + \| M(x) \nabla_x L(x, \mu) \|^2,$$

where we assume that $M(x)$ is a continuously differentiable matrix $p \times n$. A convenient choice, when the gradients of $h_i(x)$ are linearly independent, is that of taking $M(x) = \eta^{1/2} \nabla_x h(x)^T$ for some $\eta > 0$.

A function of this form can be defined as an *exact augmented Lagrangian function* because of the fact that the required correspondence with the constrained problem can be established for sufficiently small values of ε , provided that the pair (x, μ) is in some compact set $X \times L$ and that the constraints satisfy suitable

regularity conditions. In comparison with the exact penalty function we have now that the minimization must be carried out on the extended space of problem variables and multiplier estimates, but the merit function and its derivatives have a much simpler structure.

We can observe that, if we assume $M(x) = \eta^{1/2} \nabla_x h(x)^T$ and suppose that $\nabla h(x)$ as full rank we can give an alternative equivalent expression of A , by rewriting the last term as a penalty term on the difference between μ and the value of the multiplier function $\mu(x)$. In fact, we can easily verify that, if we set

$$\mu(x) = -(\nabla h(x)^T \nabla h(x))^{-1} \nabla h(x)^T \nabla f(x),$$

then we can rewrite the augmented Lagrangian in the form

$$\begin{aligned} A(x, \mu; \varepsilon) &= f(x) + \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2 + \eta \|\nabla h(x)^T \nabla h(x)(\mu(x) - \mu)\|^2 \\ &= \mu^T h(x) + \frac{1}{\varepsilon} \|h(x)\|^2 + \eta \|\nabla h(x)^T \nabla h(x)\mu + \nabla h(x)^T \nabla f(x)\|^2. \end{aligned}$$

This shows that, if we set $\mu = \mu(x)$ we obtain the exact penalty function of Fletcher.

The extension to the case of inequality constraints can be carried out by transforming the inequalities into equalities, using squared slack variables. Let us consider a problem of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \end{aligned}$$

where $g : R^n \rightarrow R^m$.

Then, we can construct an augmented Lagrangian by considering equality constraints of the form

$$g(x) + Yy = 0,$$

where $Y = \text{Diag}(y)$. In this case, the Lagrangian function is defined by:

$$\begin{aligned} \hat{A}(x, \lambda, y; \varepsilon) &= \hat{L}(x, y, \lambda) + \frac{1}{\varepsilon} \|g(x) + Yy\|^2 \\ &\quad + \eta \|\nabla(g(x) + Yy)^T \nabla \hat{L}(x, y, \lambda)\|^2, \end{aligned}$$

where

$$\hat{L}(x, y, \lambda) = f(x) + \lambda^T(g(x) + Yy),$$

$$\nabla(g(x) + Yy) = \begin{pmatrix} \nabla_x g(x) \\ 2Y \end{pmatrix},$$

$$\nabla \hat{L}(x, y, \lambda) = \begin{pmatrix} \nabla_x L(x, \lambda) \\ 2Y\lambda \end{pmatrix}.$$

The function $\hat{A}(x, \lambda, y; \varepsilon)$ can be minimized analytically with respect to the single components of y , for fixed values of x and λ .

It can be verified that for minimizing the function \hat{A} we can minimize, for each $i = 1, \dots, m$, the functions

$$\phi_i(y_i) = \lambda_i(g_i(x) + y_i^2) + \frac{1}{\varepsilon}(g_i(x) + y_i^2)^2 + \eta \left(\nabla g_i(x)^T \nabla_x L(x, \lambda) + 4y_i^2 \lambda_i \right)^2.$$

By imposing that the derivative of ϕ_i with respect to y_i is zero, letting

$$w_i(x, \lambda, \varepsilon) = g_i(x) + \frac{\varepsilon}{2} \left(\lambda_i + 8\eta\lambda_i \nabla g_i(x)^T \nabla_x L(x, \lambda) \right),$$

we get the solutions

$$y_i(x, \lambda, \varepsilon) = \left(-\frac{\min[0, w_i(x, \lambda, \varepsilon)]}{1 + 16\varepsilon\eta\lambda_i^2} \right)^{1/2}.$$

By replacing this expression into the augmented Lagrangian, we get the function

$$A(x, \lambda; \varepsilon) = f(x) + \lambda^T g(x) + \frac{1}{\varepsilon} \|g(x)\|^2 + \eta \|\nabla g(x)^T \nabla L(x, \lambda)\|^2 - \frac{1}{\varepsilon} \sum_{i=1}^m \frac{\min[0, w_i(x, \lambda, \varepsilon)]^2}{1 + 16\varepsilon\eta\lambda_i^2},$$

whose properties are analyzed in [71].

We have again that establishing the exactness of this function requires assuming that suitable regularity conditions on the constraints are satisfied and that stationary points of A are in the interior of a compact set in $R^n \times R^m$. This requirement may cause the theoretical and computational difficulties already evidenced in the case of exact penalty functions, since the level set of the augmented Lagrangian function, corresponding to some initial point and some value of the penalty parameter, may be not contained in the set where exactness is established.

In order to avoid this difficulty, augmented Lagrangian functions that include barrier terms have been introduced along similar lines to that followed in connection with exact penalty functions.

Here we briefly describe the exact augmented Lagrangian function introduced in [78], which has improved previous results. As in the preceding section we consider an open perturbation of the feasible set of the form

$$D_0 = \{x \in R^n : \sum_{i=1}^m g_i^+(x)^p < \alpha\},$$

where $p \geq 2$ and $\alpha > 0$ and we set $D = \text{Cl}(D_0)$. It is easily seen that the feasible set is contained in D_0 . We define the functions

$$a(x) = \alpha - \sum_{i=1}^m g_i^+(x)^p, \quad p(x, \lambda) = \frac{a(x)}{1 + \|\lambda\|^2}.$$

Then we can define the exact augmented Lagrangian function

$$\begin{aligned} L_a(x, \lambda; \varepsilon) = L(x, \lambda) &+ \frac{1}{2\varepsilon p(x, \lambda)} \left[\|g(x)\|^2 - \|\min\{0, g(x) + \varepsilon p(x, \lambda)\lambda\}\|^2 \right] \\ &+ \|\nabla g(x)^T \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2, \end{aligned} \tag{21.36}$$

where $\varepsilon > 0$.

We can note that, for $p(x, \lambda) = 1$ the first two terms in the r.h.s. correspond to the augmented Lagrangian function of Hestenes-Powell-Rockafellar.

The third term can be interpreted as a penalization of the equalities in the KKT conditions, through a penalty term on the difference between λ and the value of the multiplier function introduced in Sect. 21.5, expressed by $\|N(x)(\lambda(x) - \lambda)\|^2$, where

$$N(x) = \nabla g(x)^T \nabla g(x) + G^2(x),$$

and

$$\lambda(x) = -N^{-1}(x) \nabla g(x)^T \nabla f(x).$$

The term $1/p(x, \lambda)$ constitutes a barrier term that goes to infinity both when x converges to the boundary of D and when $\|\lambda\|$ goes to infinity.

The properties of L_a can be established under the following assumptions.

Assumption 21.5 (Basic Assumptions)

H1 *One of the following two conditions is satisfied*

- (a) *D is bounded*
- (b) *a feasible point \tilde{x} is available and f is coercive on D .*

H2 *For every $x \in S$ the gradients of the active constraints, $\nabla g_i(x)$, $i \in I_0(x)$ are linearly independent.*

H3 *One of the following two conditions is satisfied*

(continued)

Assumption 21.5 (continued)

- (a) at every point $x \in D$, if $\sum_{i=1}^m w_i g_i^+(x) \nabla g_i(x) = 0$, with $w > 0$, then
 $g^+(x) = 0$,
(b) a feasible point \tilde{x} is available. □

We note that H3 is satisfied if the extended Mangasarian-Fromovitz condition holds. The main properties of L_a are summarized in the propositions reported below, whose proofs can be found in [78]. In what follows we refer to the level set

$$\mathcal{L}_a(x_0, \lambda_0; \varepsilon) = \{(x, \lambda) \in D_0 \times R^m : L_a(x, \lambda; \varepsilon) \leq L_a(x_0, \lambda_0; \varepsilon)\}$$

corresponding to a point $(x_0, \lambda_0) \in D_0 \times R^m$. We suppose that, if a feasible point is available, then $x_0 \in S \subset D_0$.

A first important result guarantees that, under mild assumptions, the unconstrained minimization of L_a is well-posed for every $\varepsilon > 0$.

Proposition 21.10 (Compactness of the Level Sets) Suppose that Assumptions H1 and H2 hold. Then, for every $\varepsilon > 0$ the level set $\mathcal{L}_a(x_0, \lambda_0; \varepsilon)$ is compact. □

The preceding result guarantees both the existence of stationary points and minimizers of the penalty function and also the possibility of defining unconstrained minimization algorithms that converge to the stationary points of L_a .

The next proposition establishes the main results on the correspondence between critical points.

Proposition 21.11 (Correspondence of Critical Points) Suppose that Assumption H1 holds. Then, we have:

- (a) if $(\bar{x}, \bar{\lambda})$ is a KKT pair for the constrained problem, then, for every $\varepsilon > 0$ the pair $(\bar{x}, \bar{\lambda})$ is a stationary point of $L_a(\cdot, \cdot; \varepsilon)$;
(b) for every $\varepsilon > 0$ if $(\bar{x}, \bar{\lambda})$ is a stationary point of $L_a(\cdot, \cdot; \varepsilon)$ and

$$\max \{g(\bar{x}), -\varepsilon p(\bar{x}, \bar{\lambda})\} = 0,$$

then $(\bar{x}, \bar{\lambda})$ is a KKT pair of the constrained problem;

(continued)

Proposition 21.11 (continued)

- (c) suppose that also Assumptions H2 and H3 hold; then there exists a $\bar{\varepsilon} > 0$ such that for all $\varepsilon \in (0, \bar{\varepsilon}]$ if $(\bar{x}, \bar{\lambda}) \in \mathcal{L}_a(x_0, \lambda_0; \varepsilon)$ is a stationary point of $L_a(\cdot, \cdot; \varepsilon)$, the pair $(\bar{x}, \bar{\lambda})$ is a KKT pair of the constrained problem.

□

Now we can establish the correspondences between local and global minimizers stated below.

Proposition 21.12 (Global Minimizers of the Constrained Problem) Suppose that Assumptions H1 and H2 hold and let $x^* \in S$ be a global minimizer of the constrained problem with the associated KKT multiplier λ^* .

Then there exists $\varepsilon^* > 0$ such that, for all $\varepsilon \in (0, \varepsilon^*]$, the pair (x^*, λ^*) is a global minimum point of $L_a(\cdot, \cdot; \varepsilon)$ on $D_0 \times R^m$. □

In order to establish converse optimality results we need also the introduction of Assumption H3, in order to ensure feasibility (as we did in (c) of Proposition 21.11).

Proposition 21.13 (Minimizers of the Exact Augmented Lagrangian)

Suppose that Assumptions H1, H2 and H3 hold and let $(x_0, \lambda_0) \in D_0 \times R^m$

Then there exists $\varepsilon^* > 0$ such that, for all $\varepsilon \in (0, \varepsilon^*]$, if the point $(x^*, \lambda^*) \in \mathcal{L}_a(x_0, \lambda_0; \varepsilon)$ is a local (global) unconstrained minimum point of $L_a(\cdot, \cdot; \varepsilon)$, then x^* is a local (global) minimum point for the constrained problem and λ^* is the associated KKT multiplier. □

On the basis of the preceding results it is possible to construct an exact augmented Lagrangian algorithm, where we assume that the penalty parameter is automatically updated and $L_a(\cdot, \cdot; \varepsilon)$ is minimized with respect to (x, λ) through a globally convergent unconstrained technique for continuously differentiable functions.

The model algorithm reported below is organized in two while cycles:

- the *exterior cycle*, indexed by j , determines the value of the penalty parameter;
- the *inner cycle*, indexed by k (for each j) performs (or interrupts) the unconstrained minimization in correspondence to a fixed ε_j .

We assume that, for each fixed ε_j , the algorithm generates points (x_k, λ_k) in the level set $\mathcal{L}_a(x_0, \lambda_0; \varepsilon_j)$ such that each limit point of an infinite sequence $\{(x_k, \lambda_k)\}_j$ converges to a stationary point of $L_a(\cdot, \cdot; \varepsilon_j)$.

In the conceptual model reported below it is assumed that the algorithm terminates when we have

$$\nabla L_a(x_k, \lambda_k; \varepsilon_j) = 0, \quad \text{and} \quad \max \{g(x_k), -\varepsilon_j p(x_k, \lambda_k)\} = 0. \quad (21.37)$$

By Proposition 21.11 (b) we have that (21.37) implies that (x_k, λ_k) is a KKT pair for the constrained problem.

Exact Augmented Lagrangian Algorithm

Data $\varepsilon_0 > 0$, $(z_0, u_0) \in R^n \times R^m$ with $u_0 \geq 0$, $\sigma \in (0, 1)$.

Choose $\alpha > 0$, such that $z_0 \in D_0$, set $j = 0$ and $(x_0, \lambda_0) = (z_0, u_0)$.

If condition (21.37) is satisfied **stop**,

Do j=1, 2, ... (outer cycle)

Set $k = 0$; if $L_a(z_0, u_0; \varepsilon_j) \leq L_a(z_j, u_j; \varepsilon_j)$ set $(x_0, \lambda_0) = (z_0, u_0)$
else set $(x_0, \lambda_0) = (z_j, u_j)$.

While (inner cycle)

$$\|\nabla L_a(x_k, \lambda_k; \varepsilon_j)\| \geq \|\max \{g(x_k), -\varepsilon_j p(x_k, \lambda_k)\}\|$$

do

Compute (x_{k+1}, λ_{k+1}) using an unconstrained algorithm starting from (x_k, λ_k) and set $k = k + 1$

If condition (21.37) is satisfied **stop**

End While .

Set $\varepsilon_{j+1} = \sigma \varepsilon_j$ and $(z_{j+1}, u_{j+1}) = (x_k, \lambda_k)$.

End Do

The next proposition states the main convergence properties of the algorithm. A proof can be found in [78].

Proposition 21.14 (Convergence of the Exact Augmented Lagrangian Algorithm) *Suppose that Assumptions H1, H2 and H3 hold. Then the sequence of penalty parameters $\{\varepsilon_j\}$ is finite and either the algorithm terminates at a KKT point of the constrained problem, or every limit point of the sequence $\{(x_k, \lambda_k)\}_j$ generated in correspondence to the final value ε_j is a KKT point of the constrained problem.* \square

21.8 Exercises

- 21.1** Formulate a convergence result for the Multiplier method given in Algorithm 21.2 and give a proof of convergence, along the lines followed in the proof of Proposition 21.2.
- 21.2** Give a detailed proof of the fact that the augmented Lagrangian function L_a with inequality constraints is continuously differentiable on R^n .
- 21.3** Give a detailed proof of the fact that the exact penalty functions with shifted barriers Z is continuously differentiable on D_0 .
- 21.4** Define a computer code for solving constrained problems by choosing a penalty function or an augmented Lagrangian function and employing an unconstrained algorithm. Perform some computational experiment.

21.9 Notes and References

A basic reference text on sequential penalty functions is the book [92]. The augmented Lagrangian method has been proposed independently by Hestenes [141] and Powell [212] and later extended to inequality constrained problems by Rockafellar in [228] and in various subsequent papers. Suggested reference texts for the study of augmented Lagrangian methods are the books [15, 16, 23, 196].

Basic theoretical results and references to the literature related to non differentiable exact penalty functions can be found in [16, 74, 138].

Continuously differentiable exact penalty functions have been introduced in [97] in the equality constrained case and later extended to problems with inequality constraints in [72, 116]. Exact shifted barrier functions have been studied in [73] and in [170]. References on various modifications, extensions and computational algorithms on exact penalty methods can be found in [68, 75].

Exact augmented Lagrangians for equality constrained problems have been proposed in [70] and extended to inequality constrained problems in [71]. Exact augmented Lagrangians with barrier terms have been studied in [169] and in [78].

In this chapter we have considered exact penalty function methods, with reference to the general nonlinear programming problems. In these cases, the most advantageous computational algorithms are based on the use of the penalty function as a merit functions for globalizing local Newton-type or Quasi-Newton methods. Some example will be given in the next chapter. However, many applications to structured problems have been considered, where the computation of the multiplier functions can be performed easily and suitable specialized algorithms can be defined. Some examples are: the minimization of functions with box constraints [126], [85], the solution of minimax problems [76], the solution of quadratic problems with quadratical constraints [171], the solution of positive semidefinite relaxations of maxcut problems [130].

Chapter 22

SQP Methods



In this chapter we describe the essential features of the techniques known as *Sequential Quadratic Programming* (SQP) methods, which can be viewed as Newton-type methods for solving constrained problems. First we describe some quadratic programming problems whose KKT points, under appropriate assumptions, yield Newton-type search directions that guarantee local convergence towards a KKT point of the constrained problem at a superlinear convergence rate. Then we consider some globalization methods, based on line search techniques, employing as merit functions exact penalty functions and exact augmented Lagrangian functions.

22.1 Newton Type Methods for the Equality Constrained Case

Consider the equality constrained problem of the form

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & h(x) = 0, \end{aligned} \tag{22.1}$$

where $f : R^n \rightarrow R$ and $h : R^n \rightarrow R^p$ are twice continuously differentiable.

We can define Newton's method for solving the system of nonlinear equations obtained by imposing that the gradient of the Lagrangian function is zero. We know that if $\nabla h(x)$ has full rank and we set

$$L(x, \mu) = f(x) + \mu^T h(x),$$

then a necessary optimality condition for Problem (22.1) is that

$$\nabla L(x, \mu) = \begin{pmatrix} \nabla f(x) + \nabla h(x)\mu \\ h(x) \end{pmatrix} = 0,$$

which constitutes a nonlinear system of $n+p$ equations in the $n+p$ variables (x, μ) .

Then, if we make use of Newton's method for solving the preceding system we can attempt to construct a sequence $\{(x_k, \mu_k)\}$, starting from some initial point (x_0, μ_0) , by computing, whenever possible, the point (x_{k+1}, μ_{k+1}) that solves the linear approximation of $\nabla L(x, \mu) = 0$ at the current point. Then we can compute

$$x_{k+1} = x_k + d_k \quad \mu_{k+1} = \mu_k + s_k,$$

where the Newton's direction (d_k, s_k) solves the system:

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \mu_k) & \nabla h(x_k) \\ \nabla h(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ s \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x_k, \mu_k) \\ h(x_k) \end{pmatrix}. \quad (22.2)$$

Letting $s = \bar{\mu} - \mu_k$, the preceding system can be rewritten in the equivalent form:

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \mu_k) & \nabla h(x_k) \\ \nabla h(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ \bar{\mu} \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ h(x_k) \end{pmatrix}, \quad (22.3)$$

and the next point will be (x_{k+1}, μ_{k+1}) with $x_{k+1} = x_k + d_k$ and $\mu_{k+1} = \bar{\mu}_k$.

The same system can be derived from the quadratic programming problem

$$\begin{aligned} \min \phi(d) &= \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla_{xx}^2 L(x_k, \mu_k) d \\ h(x_k) + \nabla h(x_k)^T d &= 0, \end{aligned} \quad (22.4)$$

by writing the Lagrange multiplier rule. It can be easily verified that this yields again system (22.3), where $\mu_{k+1} = \bar{\mu}_k$ is now the Lagrange multiplier relative to the quadratic problem (22.4).

We can establish the following proposition.

Proposition 22.1 (Basic Existence Conditions)

Suppose that at a given pair (x_k, μ_k) :

- (i) the matrix $\nabla h(x_k)$ has full rank;
- (ii) the matrix $\nabla_{xx}^2 L(x_k, \mu_k)$ satisfies the second order sufficient condition

$$d^T \nabla_{xx}^2 L(x_k, \mu_k) d > 0, \quad \text{for all } d \neq 0 \text{ such that } \nabla h(x_k)^T d = 0.$$

(continued)

Proposition 22.1 (continued)
Then

- (a) *the coefficient matrix in system (22.3) is non singular and hence the system admits a unique solution $(d_k, \bar{\mu}_k)$;*
- (b) *the vector d_k is the unique global minimum point of Problem (22.4) and $\bar{\mu}_k$ is the associate Lagrange multiplier.*

Proof First we establish (a). Assume there exist $u \in R^n$, $v \in R^p$ such that we have

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \mu_k) & \nabla h(x_k) \\ \nabla h(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0.$$

We show that this implies $(u, v) = 0$, which would establish our thesis. We can write:

$$\nabla_{xx}^2 L(x_k, \mu_k)u + \nabla h(x_k)v = 0, \quad \nabla h(x_k)^T u = 0.$$

Multiplying the first equation by u^T , we obtain

$$u^T \nabla_{xx}^2 L(x_k, \mu_k)u + (\nabla h(x_k)^T u)^T v = 0,$$

and hence as $\nabla h(x_k)^T u = 0$, we get $u^T \nabla_{xx}^2 L(x_k, \mu_k)u = 0$. Therefore, assumption (ii) implies that $u = 0$, so that from the first equation we obtain $\nabla h(x_k)v = 0$ and assumption (i) implies that also $v = 0$. Thus the coefficient matrix is non singular and system (22.3) has a unique solution $(d_k, \bar{\mu}_k)$. This establishes (a).

Let now $d \in R^n$ be a feasible point of Problem (22.4) and let $z \in R^n$ be such that $d = d_k + z$. Since both d_k and d are feasible, we have

$$\nabla h(x_k)^T (d - d_k) = \nabla h(x_k)^T z = 0. \quad (22.5)$$

We can write

$$\begin{aligned} \phi(d) &= \nabla f(x_k)^T (d_k + z) + \frac{1}{2}(d_k + z)^T \nabla_{xx}^2 L(x_k, \mu_k)(d_k + z) \\ &= \phi(d_k) + z^T (\nabla_{xx}^2 L(x_k, \mu_k)d_k + \nabla f(x_k)) + \frac{1}{2}z^T \nabla_{xx}^2 L(x_k, \mu_k)z \\ &= \phi(d_k) - z^T \nabla h(x_k)\bar{\mu}_k + \frac{1}{2}z^T \nabla_{xx}^2 L(x_k, \mu_k)z \\ &= \phi(d_k) + \frac{1}{2}z^T \nabla_{xx}^2 L(x_k, \mu_k)z, \end{aligned}$$

where the last equality follows from (22.3) and (22.5). As the vector z satisfies $\nabla h(x_k)^T z = 0$ we have necessarily that $\frac{1}{2}z^T \nabla_{xx}^2 L(x_k, \mu_k)z > 0$ for $z \neq 0$, and this obviously implies that d_k is the unique global minimizer. \square

On the basis of assertion (a) of the preceding proposition, we can make use of the local convergence theory of Newton's method, provided that the algorithm is started in a neighborhood \mathcal{B} of a stationary point x^*, μ^* of the Lagrangian function $L(x, \mu)$, where the assumptions of Proposition 22.1 are satisfied. In this case, local convergence and superlinear convergence towards (x^*, μ^*) can be established and moreover, if $\nabla^2 L(x, \mu)$ is Lipschitz continuous in \mathcal{B} , also a Q-quadratic convergence rate can be obtained.

In alternative to this primal-dual version of Newton's method, we can also define a primal version if we introduce a continuous *multiplier function* $\mu(x)$ such that $\mu(x^*) = \mu^*$ at x^* . In this case, given x_k , we can first compute $\mu_k = \mu(x_k)$ and then the search direction d_k by solving the linear system

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \mu_k) & \nabla h(x_k) \\ \nabla h^T(x_k) & 0 \end{pmatrix} \begin{pmatrix} d \\ s \end{pmatrix} = - \begin{pmatrix} \nabla_x L(x_k, \mu_k) \\ h(x_k) \end{pmatrix}. \quad (22.6)$$

Thus we can compute $x_{k+1} = x_k + d_k$. Now, the component s_k of the solution is not used for defining μ_{k+1} , which is rather defined by $\mu_{k+1} = \mu(x_{k+1})$.

22.2 Extension to Inequality Constrained Problems

Consider now the problem with equality and inequality constraints, that is

$$\begin{aligned} \min f(x) \\ g(x) \leq 0, \quad h(x) = 0, \end{aligned} \quad (22.7)$$

where $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$ are twice continuously differentiable.

We can extend the preceding results by considering the quadratic programming problem

$$\begin{aligned} \min_d \quad & \nabla f^T(x)d + \frac{1}{2}d^T \nabla_{xx}^2 L(x, \lambda, \mu)d \\ \text{subject to} \quad & g(x) + \nabla g(x)^T d \leq 0, \\ & h(x) + \nabla h(x)^T d = 0, \end{aligned} \quad (22.8)$$

where now also the inequality constraints have been linearized. It can be verified that the KKT optimality condition for this problem correspond to a linear approximation of the KKT conditions for the constrained problem.

Given a point (x_k, λ_k, μ_k) a “local” algorithm can be defined by finding (when possible) a KKT point (d_k, ρ_k, η_k) of Problem (22.8), where ρ_k, η_k are the KKT multipliers associated to inequalities and equalities in Problem (22.8). Then we can set

$$x_{k+1} = x_k + d_k, \quad \lambda_{k+1} = \rho_k, \quad \mu_{k+1} = \eta_k.$$

In the general case, establishing the local convergence of an algorithm based on SQP is more difficult than in the equality constrained case. However, under appropriate assumptions, it can be established that the algorithm is well defined and that the primal-dual convergence rate of (x_k, λ_k, μ_k) is superlinear (and quadratic under Lipschitz continuity assumptions on the Hessian matrices of the problem functions). A superlinear convergence rate of $\{x_k\}$ can also be demonstrated.¹ We refer, for instance, to [226] and [112].

The Newton-type methods considered above are based on the solution of an inequality constrained quadratic programming problem and this approach is referred to as IQP (*inequality-constrained QP*). The main drawback is the need of solving an inequality constrained quadratic programming problem at each step.

An alternative approach is that of choosing at each iteration a set of constraints, which are estimated to be active at the solution and then solving an equality constrained quadratic programming problem where the remaining constraints are provisionally ignored. This approach is referred to as EQP (*equality-constrained QP*) and will be considered in the next section.

22.3 EQP Methods

To simplify notation we refer to nonlinear programming problems with inequality constraints, of the form:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0, \end{aligned} \tag{22.9}$$

where $f : R^n \rightarrow R$ and $g : R^n \rightarrow R^m$, $h : R^n \rightarrow R^p$ are twice continuously differentiable.

Suppose that $(\bar{x}, \bar{\lambda})$ is a KKT point of Problem (22.9). An EQP method can be constructed, in a neighborhood of the point \bar{x} , by defining at each step an index set $A_k \subseteq \{1, 2, \dots, m\}$ of the constraints that are estimated to be active at \bar{x} .

Then, we can define the iteration

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

where, after reordering, d_k is given by the solution of a linear system of the form

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \lambda_k) & \nabla g_{A_k}(x_k) \\ \nabla g_{A_k}(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ z \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_{A_k}(x_k) \end{pmatrix}, \tag{22.10}$$

and λ_k is an estimate of the KKT multiplier $\bar{\lambda}$.

¹ We recall that, in general, the superlinear or quadratic convergence rate of a sequence formed by a pair of vectors (u_k, v_k) does not imply the superlinear or quadratic convergence of the sequence formed with a single vector u_k or v_k .

We will refer, in particular, to the approach proposed in [86] where, under suitable assumptions, both the set A_k and the multiplier estimate λ_k are constructed by employing a *multiplier function*, that is a continuously differentiable function $\lambda : R^n \rightarrow R^m$, defined in a neighborhood \mathcal{B} of \bar{x} , such that $\lambda(\bar{x}) = \bar{\lambda}$.

We suppose that the following assumptions are satisfied.

Assumption 22.1 (Basic Assumptions)

- (i) *The linear independence constraints qualification (LICQ) holds at \bar{x} .*
- (ii) *The (semi) strong second order sufficiency conditions holds, that is*

$$d^T \nabla_x^2 L(\bar{x}, \bar{\lambda}) d > 0,$$

for all $d \in R^n$, $d \neq 0$ such that $\nabla g_i(\bar{x})^T d = 0$, $i \in I_0^+(x^)$, where*

$$I_0^+(x^*) = \{i \in I_0(x^*) : \lambda_i^* > 0\}$$

and

$$I_0(x^*) = \{i : g_i(x^*) = 0\}.$$

□

Under the assumption stated, we can consider, in particular, the multiplier function (already introduced) proposed in [170] and defined by

$$\lambda(x) = -N^{-1}(x) \nabla g(x)^T \nabla f(x), \quad (22.11)$$

where $N(x) = \nabla g(x)^T \nabla g(x) + \gamma_1 G^2(x) + \gamma_2 r(x) I_m$, with $\gamma_1 > 0$, $\gamma_2 \geq 0$, and $r(x) = \sum_{i=1}^m (g_i^+(x))^3$.

We recall that in a neighborhood of \bar{x} the function is well defined (even if $\gamma_2 = 0$) and continuously differentiable. At the KKT point $(\bar{x}, \bar{\lambda})$ we have, by construction, that $\lambda(\bar{x}) = \bar{\lambda}$.

Using a multiplier function (possibly different from that defined above), we can introduce the following estimate at x of $I_0(\bar{x})$,

$$A(x) = \{i : g_i(x) \geq -\rho(x)\lambda_i(x)\}, \quad (22.12)$$

where $\rho(x)$ is a continuous function defined in a neighborhood of \bar{x} and such that we have $\rho_2 \geq \rho(x) \geq \rho_1$ for some $\rho_2 \geq \rho_1 > 0$.

Then, it easily seen that there exists a neighborhood \mathcal{B} of \bar{x} such that in this neighborhood we have

$$I_0^+(\bar{x}) \subseteq A(x) \subseteq I_0(\bar{x}). \quad (22.13)$$

In fact, suppose that the neighborhood is sufficiently small to have that the multiplier function is well defined and that, for all x in this neighborhood we have

$$\begin{aligned} \lambda_i(x) &> 0 \quad \text{if } \bar{\lambda}_i = \lambda_i(\bar{x}) > 0, \\ \rho_2 &\geq \rho(x) \geq \rho_1 > 0, \\ g_i(x) &< 0 \quad \text{if } g_i(\bar{x}) < 0. \end{aligned}$$

Then, if $i \in I_0^+(\bar{x})$, we must have $\lambda_i(\bar{x}) > 0$ and $g_i(\bar{x}) = 0$, so that

$$g_i(\bar{x}) + \rho(\bar{x})\lambda_i(\bar{x}) = \rho(\bar{x})\lambda_i(\bar{x}) > 0,$$

which implies, by continuity, that, for $x \in \mathcal{B}$ we have $g_i(x) + \rho(x)\lambda_i(x) > 0$, so that $i \in A(x)$, and this establishes the first inclusion. Now, if $i \notin I_0(\bar{x})$, we have $g_i(\bar{x}) < 0$ and $\bar{\lambda}_i = \lambda_i(\bar{x}) = 0$, so that $g_i(\bar{x}) + \rho(\bar{x})\lambda_i(\bar{x}) = g_i(\bar{x}) < 0$, and hence, by continuity, we have that $i \notin A(x)$ for $x \in \mathcal{B}$. This implies that the second inclusion must hold.

We can now define the Newton type algorithm $x_{k+1} = x_k + d_k$, where d_k is obtained by solving the linear system

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \lambda(x_k)) & \nabla g_{A(x_k)}(x_k) \\ \nabla g_{A(x_k)}(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ z \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_{A(x_k)}(x_k) \end{pmatrix}. \quad (22.14)$$

When $x_k \in \mathcal{B}$, under Assumption 22.1, we can consider the above system equivalent to system (22.4), with $h \equiv g_{A(x_k)}$, so that it follows from Proposition 22.1 that d_k is well defined.

From a theoretical point of view, it can be shown [86] that, if the problem functions are twice continuously differentiable with Lipschitz continuous Hessian matrices and Assumption 22.1 holds, then there exists a neighborhood $\mathcal{B}_1 \subseteq \mathcal{B}$ of \bar{x} such that, if $x_0 \in \mathcal{B}_1$, the algorithm described above is well defined and $\{x_k\}$ converges Q-superlinearly to \bar{x} . Moreover, if the multiplier function is Lipschitz continuous at \bar{x} , the convergence is Q-quadratic. In particular, the multiplier function defined in (22.11) satisfies this condition.

From a computational point of view, the algorithm requires at each step

- the computation of $\lambda(x_k)$, which typically consists in the solution of a $m \times m$ linear system (this is the case, in particular, if the multiplier function (22.11) is employed);
- the solution of the system (22.14).

If we tolerate a deterioration in the convergence rate, the computational cost can be reduced, by computing the approximation λ_k from the solution obtained at the preceding step and thus avoiding the solution of a linear system.

More specifically, given an initial estimate $x_0 \in \mathcal{B}_1$, $\lambda_0 \in R^m$ we can define the set

$$A_k = \{i : g_i(x_k) \geq -\rho \lambda_{ik}\}, \quad \rho > 0. \quad (22.15)$$

Then, we can consider the iteration $x_{k+1} = x_k + d_k$, where d_k is obtained from the solution (d_k, z_k) of the linear system

$$\begin{pmatrix} \nabla_{xx}^2 L(x_k, \lambda_k) & \nabla g_{A_k}(x_k) \\ \nabla g_{A_k}(x_k)^T & 0 \end{pmatrix} \begin{pmatrix} d \\ z \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ g_{A_k}(x_k) \end{pmatrix}, \quad (22.16)$$

and we set

$$\lambda_{i(k+1)} = z_{ik} \text{ if } i \in A_k, \quad \lambda_{i(k+1)} = 0 \text{ if } i \notin A_k.$$

Under the same assumptions introduced in connection to the iteration (22.14), it can be shown that there exists a neighborhood V of $(\bar{x}, \bar{\lambda})$ such that, if $(x_0, \lambda_0) \in V$, the algorithm is well defined and $\{(x_k, \lambda_k)\}$ converges to $(\bar{x}, \bar{\lambda})$ with a quadratic convergence rate. However, in this case it is only possible to establish the superlinear convergence rate of $\{x_k\}$.

22.4 Quasi-Newton Modification

In many cases the second order derivatives may be not easily available. Moreover, it may happen that the matrix $\nabla_{xx}^2 L$ must be modified in order to guarantee the existence of solutions of the quadratic programming subproblem.

A positive definite approximation to the Hessian of the Lagrangian can be based on a suitable modification of some Quasi-Newton formula. In principle, if B_k is a symmetric positive definite matrix, the updated matrix B_{k+1} , computed through the BFGS formula for approximating $\nabla_{xx}^2 L$ at the next step, should be given by

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k}, \quad (22.17)$$

where

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_k).$$

We know, however that B_{k+1} would remain positive definite if and only if $y_k^T s_k > 0$. In the unconstrained case this condition is satisfied through the adoption of a line search along the Quasi-Newton direction. In this case no line search is performed on L for constructing the quadratic programming subproblem and therefore it has been suggested by Powell [216] to modify, when required, the expression of y_k in the BFGS formula in order to guarantee that the updated matrix is positive definite. In particular, Powell's update consists in replacing in formula (22.17) the vector y_k with the vector

$$\hat{y}_k = \begin{cases} y_k & \text{if } s_k^T y_k \geq 0.2 s_k^T B_k s_k \\ \theta_k y_k + (1 - \theta_k) B_k s_k & \text{if } s_k^T y_k < 0.2 s_k^T B_k s_k, \end{cases} \quad (22.18)$$

where

$$\theta_k = \frac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - s_k^T y_k}.$$

It can be easily verified that, in all cases, we have $s_k^T y_k \geq 0.2 s_k^T B_k s_k > 0$.

The modification defined above is known as *damped Quasi-Newton method* and has been successfully employed also in the unconstrained case [2]. However, other possible modifications can be considered (see, e.g., [196]).

22.5 Solution of the Quadratic Programming Subproblem

A computational implementation of SQP methods for solving constrained optimization problems must take into account several potential difficulties. In the definition of the local algorithm, a first point is that the quadratic programming problem can be infeasible, that is the linearized system may not have a solution. In this case we can modify the linearized system by introducing artificial variables. In the general case, we can replace problem (22.8) with the problem

$$\begin{aligned} \min \quad & \nabla f^T(x)d + \frac{1}{2}d^T \nabla_{xx}^2 L(x, \lambda, \mu)d + \tau \sum_{i=1}^p (u_i + v_i) + \tau w \\ \text{subject to} \quad & g(x) + \nabla g_i(x)^T d \leq w, i = 1, \dots, m \\ & h_i(x) + \nabla h_i(x)^T d = u_i - v_i, \\ & u_i \geq 0, v_i \geq 0, i = 1, \dots, p, w \geq 0. \end{aligned} \quad (22.19)$$

where w, u_i, v_i are artificial variables, and τ is a positive coefficient. Under appropriate assumptions, it can be shown that, for sufficiently large values of τ , this problem is equivalent to problem (22.8) if the QP problem is feasible, and in this case the artificial variables w, u_i, v_i will be zero at the optimal solution of problem (22.19).

Even if the QP problem is feasible, this problem may not admit an optimal solution and the objective function can be unbounded below. To prevent this situation a modification of the Hessian matrix could be required. In particular, the Hessian of the Lagrangian can be replaced, at each step k , with a positive definite matrix M_k , obtained, for instance, from modified Quasi-Newton formulae.

In particular, in the inequality constrained case, if we adopt a IQP model, we can refer, at each k , to the local subproblem:

$$\begin{aligned} \min \quad & \nabla f^T(x_k) d + \frac{1}{2} d^T M_k d + \tau w \\ \text{s.t.} \quad & g_i(x_k) + \nabla g_i(x_k)^T d \leq w, i = 1, \dots, m \\ & w \geq 0. \end{aligned} \tag{22.20}$$

It easily seen that this problem is feasible, since, for instance, we can choose arbitrarily $d_0 \in R^n$ and then

$$w_0 = \max \left\{ 0, \max_{i=1, \dots, m} \{g_i(x_k) + \nabla g_i(x_k)^T d_0\} \right\}.$$

Moreover, if M_k is positive definite the objective function is coercive (and convex) on the feasible set and hence it admits an optimal solution.

However, the globalization strategy can also require that the quadratic programming subproblem is abandoned in case of difficulties, by computing the search direction with a different criterion.

22.6 Globalization Strategies

Globalization methods have the objective of enforcing convergence from arbitrary starting points. The typical strategy is based on the introduction of a *merit function* and we can distinguish:

- *linesearch methods*: a line search related to the merit function is performed along a search direction computed by solving (approximately) the local quadratic programming problem;
- *trust region methods* the merit function is used for monitoring the current step and for controlling the size of the trust region radius.

In the sequel we will refer only to line search globalization; algorithms based on trust region are considered in the references quoted at the end of this chapter.

Many different merit functions have been employed, and, in particular: nondifferentiable and continuously differentiable penalty functions, augmented Lagrangian methods, exact augmented Lagrangian methods.

In order to simplify notation we will refer, unless otherwise stated, to problems with inequality constraints, that is to problems of the form:

$$\begin{aligned} \min & f(x) \\ g(x) &\leq 0. \end{aligned} \tag{22.21}$$

22.6.1 Nondifferentiable Merit Functions

The non differentiable merit functions typically employed for the globalization of SQP methods are the ℓ_1 penalty function

$$J_1(x; \varepsilon) = f(x) + 1/\varepsilon \|g^+(x)\|_1 \equiv f(x) + 1/\varepsilon \sum_{i=1}^m g_i^+(x),$$

and the ℓ_∞ penalty function

$$J_\infty(x; \varepsilon) = f(x) + 1/\varepsilon \|g^+(x)\|_\infty \equiv f(x) + 1/\varepsilon \max\{g_1^+(x), \dots, g_m^+(x)\},$$

where $g_i^+(x) \equiv \max\{0, g_i(x)\}$.

We will confine ourselves to illustrate the globalization scheme based on the function J_1 defined above, assuming that, at each step k the search direction d_k is computed by solving the quadratic programming problem in (d, w) :

$$\begin{aligned} \min & \nabla f^T(x_k)d + \frac{1}{2}d^T M_k d + \tau w \\ g(x_k) + \nabla g(x_k)^T d &\leq we, \quad w \geq 0, \end{aligned} \tag{22.22}$$

where $\tau > 0$, $e \in R^m$ is the unit vector, and we assume that the symmetric matrices M_k remain in a compact set of symmetric positive definite matrices for all k .

The globalization algorithm consists essentially in performing at each step a line search (such as Armijo' line search) along the direction d_k obtained by solving problem (22.22). Before defining more precisely a computational scheme, we show that, for sufficiently small values of ε , the direction d_k is a descent direction for J_1 when x_k is not a critical point of a feasible constrained problem.

First we observe that, if d_k, w_k is an optimal solution of problem (22.22) then the KKT conditions must be satisfied (as the constraints are linear in d, w) and hence

there must exist multipliers $u_k = (u_{1k}, \dots, u_{mk})^T \in R^m$ and $v_k \in R$ such that

- (i) $\nabla f(x_k) + M_k d_k + \nabla g(x_k) u_k = 0$
 - (ii) $\tau - u_k^T e - v_k = 0$
 - (iii) $u_k^T (g(x_k) + \nabla g(x_k)^T d_k - w_k e) = 0,$
 - (iv) $v_k w_k = 0$
 - (v) $u_k \geq 0, \quad v_k \geq 0$
 - (vi) $g(x_k) + \nabla g(x_k)^T d_k - e w_k \leq 0,$
 - (vii) $w_k \geq 0.$
- (22.23)

By (22.23) we can write

$$\begin{aligned} \nabla f(x_k)^T d_k &= -d_k^T M_k d_k - u_k^T (\nabla g(x_k)^T d_k) \\ &= -d_k^T M_k d_k + u_k^T (g(x_k) - e w_k) \\ &= -d_k^T M_k d_k + u_k^T g(x_k) - \tau w_k \end{aligned} \quad (22.24)$$

where we have employed the identity

$$w_k u_k^T e = \tau w_k,$$

which follows from (ii) and (iv). Now, using again (22.23), we can write:

$$\begin{aligned} \|g^+(x_k + t d_k)\|_1 &= \sum_{i=1}^m \max[0, g_i(x_k + t d_k)] = \sum_{i=1}^m \max[0, g_i(x_k) \\ &\quad + t \nabla g_i(x_k)^T d_k + \sigma_i(x_k, t d_k)] \\ &= \sum_{i=1}^m \max[0, g_i(x_k)(1-t) + t(g_i(x_k) \\ &\quad + \nabla g_i(x_k)^T d_k) + \sigma_i(x_k, t d_k)] \\ &\leq (1-t) \|g^+(x_k)\|_1 + t m w_k + \sum_{i=1}^m \max[0, \sigma_i(x_k, t d_k)], \end{aligned} \quad (22.25)$$

where $\lim_{t \rightarrow 0^+} \sigma_i(x_k, t d_k)/t = 0$. Therefore, we have

$$\lim_{t \rightarrow 0^+} \frac{\|g^+(x_k + t d_k)\|_1 - \|g^+(x_k)\|_1}{t} \leq -\|g^+(x_k)\|_1 + m w_k. \quad (22.26)$$

The directional derivative of the merit function at x_k along d_k is given by

$$D J_1(x_k, d_k; \varepsilon) = \nabla f(x_k)^T d_k + \frac{1}{\varepsilon} \lim_{t \rightarrow 0^+} \frac{\|g^+(x_k + t d_k)\|_1 - \|g^+(x_k)\|_1}{t}. \quad (22.27)$$

Then, by (22.24) and (22.26) we obtain

$$D J_1(x_k, d_k : \varepsilon) \leq -d_k^T M_k d_k + u_k^T g(x_k) - \tau w_k + \frac{1}{\varepsilon} (-\|g^+(x_k)\| + m w_k). \quad (22.28)$$

As $u_k \geq 0$ we have

$$u_k^T g(x_k) \leq u_k^T g^+(x_k) \leq \max_i \{u_{ik}\} \|g^+(x_k)\|_1.$$

Then, if we assume that $d_k \neq 0$ and that

$$\max_i \{u_{ik}\} \leq 1/\varepsilon, \quad \tau \geq m/\varepsilon,$$

we have

$$D J_1(x_k; d_k) \leq -d_k^T M_k d_k < 0, \quad (22.29)$$

and hence d_k is a descent direction for the merit function J_1 .

We note also that, if $d_k = 0$ and $w_k = 0$, the point x_k with the associate multiplier u_k , satisfies the KKT conditions for the constrained problem.

A conceptual algorithm model of a linesearch-based globalization algorithm is given in the following scheme. To simplify our discussion, we assume that the quadratic programming problem (22.22) is feasible with $w = 0$, (which implies a regularity condition on the feasible set of the original constrained problem).

Linesearch-Based Globalization Algorithm for SQP (Feasible QP Subproblem)

Initial step Choose $x_0 \in R^n$, $\lambda_0 \in R^n$, ε such that $1/\varepsilon > \|\lambda_0\|_\infty$, and set $k = 0$.

For k=0,1,2,...

1. If x_k, λ_k is a KKT point terminate.
2. Choose a definite positive matrix M_k .
3. Determine a KKT point d_k, u_k of the quadratic programming problem (22.22). Terminate if $d_k = 0$ (the point x_k, u_k is a KKT point).
4. Modify, if needed, the penalty parameter ε .
5. If $d_k \neq 0$ determine $x_{k+1} = x_k + \alpha_k d_k$ through a linesearch in order to guarantee a sufficient decrease of the merit function $J_1(x; \varepsilon)$ and update the multiplier estimate, by setting, for instance, $\lambda_{k+1} = u_k$.

End For □

A linesearch algorithm can be based, for instance, on an Armijo-type algorithm, where the sufficient decrease of the objective function consists in the satisfaction of the acceptance rule

$$J_1(x_k + \alpha_k d_k; \varepsilon_k) \leq J_1(x_k; \varepsilon_k) - \gamma \alpha_k d_k^T M_k d_k,$$

where $\gamma \in (0, 1/2)$ and the initial tentative step-size is the unit step-size.

Recalling (22.29), it can be easily verified that, under the assumption that $1/\varepsilon_k > \|u_k\|_\infty$ and that $d_k \neq 0$, the acceptance criterion is satisfied for sufficiently small values of α .

Many different implementations of the globalization algorithm outlined above have been proposed, which differ in the line search rules, in the criterion used for updating the multiplier, in the definition of M_k , in the rules adopted for choosing and updating the penalty parameter.

The main limitation of globalization algorithms based on non differentiable penalty functions is the fact that the line search may not accept the unit step-size even if the limit point satisfies the assumptions that should guarantee a superlinear convergence rate when $\alpha = 1$. This phenomenon is known as *Maratos effect*, since some examples given by N. Maratos [181] demonstrate that in some cases there is no neighborhood of a limit point where the unit step-size can be accepted.

Several modifications of the basic scheme have been proposed in order to overcome this limitation, and in particular:

- (i) introduction of a second order correcting step
- (ii) use of a non differentiable augmented Lagrangian
- (iii) application of non monotone acceptance rules for the step-size along d_k .

For an illustration of points (i) and (ii) and many relevant references we refer to [112]; point (iii) will be analyzed in Chap. 24.

An alternative approach that will be introduced in the next subsection is the definition of linesearch-based globalization algorithms employing continuously differentiable merit functions, such as exact penalty functions or augmented Lagrangian methods.

22.6.2 Smooth Merit Functions

The introduction of smooth merit functions for globalizing SQP techniques can prevent the occurrence of the Maratos effect and also allow us to employ algorithms for continuously differentiable unconstrained problems whenever the solution of the quadratic programming subproblems is unsatisfactory. In particular, the continuously differentiable exact penalty functions introduced in Chap. 21, the augmented Lagrangian functions and the exact augmented Lagrangian functions have been employed as merit functions in connection with SQP methods.

In the sequel we will confine ourselves to illustrate the essential properties of an algorithm based on the exact augmented Lagrangian function, already introduced in Chap. 21, which contains shifted barrier terms on the constraints, with respect to primal variables, and it is coercive with respect to dual variables. This function can be used as merit function, in connection with the EQP subproblem proposed in [86] and already described in Sect. 22.3. We refer, in particular, to the algorithm proposed in [77], which is globally and superlinearly convergent under reasonable assumptions.

Here we assume that the problem functions f and g are three times continuously differentiable, but third order derivatives are never evaluated.

As in Chap. 21, we consider an open perturbation of the feasible set of the form

$$D_0 = \{x \in R^n : \sum_{i=1}^m g_i^+(x)^3 < \alpha\},$$

where $\alpha > 0$ and we set $D = \text{Cl}(D_0)$. Then we define the functions

$$a(x) = \alpha - \sum_{i=1}^m g_i^+(x)^3, \quad p(x, \lambda) = \frac{a(x)}{1 + \|\lambda\|^2}.$$

The exact augmented Lagrangian function is given by

$$\begin{aligned} L_a(x, \lambda; \varepsilon) = L(x, \lambda) + \frac{1}{2\varepsilon p(x, \lambda)} & \left[\|g(x)\|^2 - \|\min\{0, g(x) + \varepsilon p(x, \lambda)\}\|^2 \right] \\ & + \|\nabla g(x)^T \nabla_x L(x, \lambda) + G(x)^2 \lambda\|^2, \end{aligned} \tag{22.30}$$

where $\varepsilon > 0$. We suppose that the conditions given in Assumption 21.5 of Chap. 21 are satisfied, so that the properties of exactness reported in that chapter must hold.

In order to define a globalization algorithm that enforces convergence from poor starting points, while preserving the ultimate superlinear convergence rate of EQP methods, we must show that:

- (i) at each step we can determine a search direction in primal-dual space that guarantees global convergence towards stationary points of L_a ;
- (ii) the penalty parameter ε , in a finite number of iterations, should become sufficiently small to have that the limit points are also KKT points of the constrained problem;
- (iii) there must exist a neighborhood of a KKT point satisfying the assumption stated in Sect. 22.3, where the algorithm does not modify the solution of the quadratic programming subproblem.

Condition (i) can be satisfied, as the function L_a is continuously differentiable and for each $\varepsilon > 0$ the level set

$$\mathcal{L}_a(x_0, \lambda_0; \varepsilon) = \{(x, \lambda) \in D_0 \times R^m : L_a(x, \lambda; \varepsilon) \leq L_a(x_0, \lambda_0; \varepsilon)\}$$

is compact. Therefore gradient-related search directions can be defined such that the limit points of the iterates generated through an Armijo-type line search are stationary points of $L_a(., .; \varepsilon)$ in $\mathcal{L}_a(x_0, \lambda_0; \varepsilon)$. In particular, we can check whether the solution of the quadratic programming subproblem satisfies some sufficient convergence condition for unconstrained minimization and switch to the negative gradient direction $-\nabla L_a$ in case of failure.

Condition (ii) can be satisfied because of the properties of exactness of the merit function, which have been established under the assumptions stated (see Chap. 21). As the threshold value $\bar{\varepsilon}$ of the penalty parameter is not known, this requires automatic adjustment rules that guarantee in a finite number of steps that $\varepsilon \leq \bar{\varepsilon}$.

As regards condition (iii), it can be shown that, for sufficiently small values of ε , there exists a neighborhood of a KKT point where the direction computed through the local algorithm is a good descent direction and the Armijo-type line search accepts the unit step-size, so that the algorithm retains a superlinear convergence rate. This is possible even if the quadratic programming subproblem is solved only approximately, by employing truncated procedures, provided that the precision of the solution process can be controlled.

Although the general ideas at the basis of the overall algorithm are quite clear, the detailed description and the convergence proof would require too much space and hence the interested reader is referred to the literature.

22.7 Notes and References

Useful references for the SQP methods are, in particular, the books [16, 196] and the work [112], which contain also extensive bibliographies and historical notes on Newton-type SQP methods.

Differentiable exact penalty functions and augmented Lagrangian functions have been used typically as merit functions for globalizing SQP algorithms. In particular, Fletcher's exact penalty function has been employed in a SQP algorithm for equality constrained problems in [220]. The extension of this approach to problems with inequality constraints has been given in [69].

The use of exact augmented Lagrangian methods in SQP algorithms has been considered in [15]. The algorithm described in the last section, which makes use of an exact augmented Lagrangian with shifted barrier terms, is described in detail in [77].

Chapter 23

Introduction to Interior Point Methods



In this chapter we give a short introduction to interior point methods (IPMs). We start from early results given in the 1960s on barrier methods and then we outline some developments of this field in relation to Linear Programming (LP), after Karmarkar's paper on a polynomial time interior point method. In particular, we describe primal-dual algorithms for LP, which are the best known interior point techniques. Finally we report some extensions to nonlinear programming.

23.1 Basic Definitions and Barrier Methods

Let us consider a mathematical programming problem of the form

$$\begin{aligned} & \min f(x) \\ & x \in H, \quad g(x) \leq 0, \end{aligned} \tag{23.1}$$

where $H \subseteq R^n$ is a closed set and the functions $f : R^n \rightarrow R$, $g : R^n \rightarrow R^m$ are assumed to be at least continuous. We denote by $\mathcal{F} = \{x \in H : g(x) \leq 0\}$ the feasible set and we define the set

$$D = \{x \in R^n : g(x) < 0\},$$

that is the set (possibly empty) where the inequality constraints $g(x) \leq 0$ are satisfied as strict inequalities. In particular, consider an LP problem in *standard form*

$$\begin{aligned} & \min c^T x \\ & Ax = b, \quad x \geq 0, \end{aligned} \tag{23.2}$$

where $x, c \in R^n$, $A(m \times n)$ and $b \in R^m$. We can define the sets

$$H = \{x \in R^n : Ax - b = 0\} \quad D = \{x \in R^n : x > 0\},$$

$$\mathcal{F} = \{x \in R^n : Ax - b = 0, x \geq 0\}.$$

We speak, in general, of *interior point methods* with reference to techniques such that a solution of the constrained problem (23.1) is approximated through one of the following sequences:

- a sequence of points in D (*infeasible methods*);
- a sequence of points in $H \cap D$ (*feasible methods*).

In both cases, the essence of an interior point method is that of remaining in D , thus maintaining always strictly satisfied the inequality constraints $g(x) \leq 0$.

We consider here the case of feasible methods for inequality constrained problems, such that $H = R^n$, D is non empty and a point $x_0 \in D$ is available. In this case the motivation of interior point methods is essentially that of employing directly solution techniques based on unconstrained minimization algorithms, without introducing penalty terms on the constraints.

IPMs can be based on the concept of barrier function, defined below.

Definition 23.1 (Barrier Function) Let $g : R^n \rightarrow R^m$ and $D = \{x \in R^n : g(x) < 0\} \neq \emptyset$. We say that a continuous function $f_B : D \rightarrow R$ is a barrier function for D if for every sequence $\{x_k\}$ of points in D converging to some $\bar{x} \in R^n$, such that $g_i(\bar{x}) = 0$ for at least one i , we have $\lim_{k \rightarrow \infty} f_B(x_k) = +\infty$.

□

The best known barrier functions are those already introduced in the first works in the 1950s on interior point methods and, in particular:

- the *logarithmic barrier function*:

$$f_B(x) = - \sum_{i=1}^m \ln(-g_i(x));$$

- the *inverse barrier function*:

$$f_B(x) = - \sum_{i=1}^m \frac{1}{g_i(x)}.$$

Barrier methods require also the introduction of a merit function (typically called *potential function*) of the form:

$$\Psi(x; r) = f(x) + rf_B(x),$$

where f_B is a barrier function and $r > 0$.

We refer, for simplicity, to the study of global minimizers, but the result can be extended, under mild assumptions, to the case of local minimizers.

We can define the following conceptual algorithm model, where we consider the case of a feasible algorithm, where H is a closed set and f_B is a barrier function on D .

Barrier Method

Data Choose $r^0 > 0$, $x_0 \in H \cap D$ and set $k = 0$.

For k=0,1,...

Compute a point $x^{k+1} \in \arg \min_{x \in H \cap D} \Psi(x; r_k)$;

Choose $r^{k+1} < r^k$.

End for.

Now, along the lines followed in [16], we state a convergence result for problem (23.1) under the following assumptions.

Assumption 23.1

- (i) The set H is closed, the set $H \cap D = \{x \in H : g(x) < 0\}$ is non empty and the feasible set $\mathcal{F} = \{x \in H : g(x) \leq 0\}$ is bounded;
- (ii) the functions f and g are continuous on \mathcal{F} ;
- (iii) for every $x \in \mathcal{F}$ and $\rho > 0$ there exists $y \in H \cap D$ such that $\|x - y\| < \rho$;
- (iv) the function f_B is a barrier function on D in the sense of Definition 23.1.

Proposition 23.1 (Global Convergence of a Barrier Method)

Suppose that Assumption 23.1 holds and that $\{r_k\}$ is a sequence of positive numbers converging to zero. Then:

- (i) for every $r_k > 0$ the potential function $\Psi(x; r_k)$ has a global minimizer $x_k \in H \cap D$;
- (ii) the sequence $\{x_k\}$ has limit points in \mathcal{F} and every limit point is a global minimizer of the constrained problem.

Proof As $H \cap D$ is non empty we can find a point $\hat{y} \in H \cap D$ and we can consider, in correspondence to a fixed value of r , the non empty level set:

$$\mathcal{L}_r(\hat{\alpha}) = \{y \in H \cap D : \Psi(y; r) \leq \hat{\alpha}\}, \quad \text{where } \hat{\alpha} = \Psi(\hat{y}; r).$$

We show that $\mathcal{L}_r(\hat{\alpha})$ is compact. As \mathcal{F} is bounded, we must only show that $\mathcal{L}_r(\hat{\alpha})$ is closed. Let $\{y_k\}$ be a sequence in $\mathcal{L}_r(\hat{\alpha})$ converging to a limit point \bar{y} . As H is closed, we have $\bar{y} \in H$ and by continuity of g , we have, $g(\bar{y}) \leq 0$. Reasoning by contradiction, we can assume that $\bar{y} \notin D$; this implies that there must exist an index i such that $g_i(\bar{y}) = 0$. As f_B is a barrier function, we have

$$\lim_{k \rightarrow \infty} f_B(y_k) = +\infty$$

and hence, as f is bounded on \mathcal{F} and $r > 0$, we have

$$\lim_{k \rightarrow \infty} \Psi(y_k; r) = +\infty,$$

which contradicts, for sufficiently large k , the assumption that $y_k \in \mathcal{L}_r(\hat{\alpha})$. It follows that $\mathcal{L}_r(\hat{\alpha})$ is compact and hence, by continuity of f assertion (i) follows from known results.

Let now $\{x_k\}$ be a sequence of minimizers of $\Psi(x; r_k)$. These points belong, by construction, to the compact set \mathcal{F} and hence there exists a subsequence $\{x_k\}_K$ converging to some $x^* \in \mathcal{F}$. If $x^* \in D$, as $r_k \rightarrow 0$, we have

$$\lim_{k \rightarrow \infty, k \in K} r_k f_B(x_k) = 0.$$

If $x^* \notin D$, as f_B is a barrier function, we have

$$\lim_{k \rightarrow \infty, k \in K} f_B(x_k) = +\infty,$$

and hence, we have, in all cases,

$$\liminf_{k \rightarrow \infty, k \in K} r_k f_B(x_k) \geq 0.$$

Therefore we can write

$$\liminf_{k \rightarrow \infty, k \in K} (f(x_k) + r_k f_B(x_k)) = f(x^*) + \liminf_{k \rightarrow \infty, k \in K} r_k f_B(x_k) \geq f(x^*). \quad (23.3)$$

Now, by contradiction, let us assume that x^* is not a global minimizer of the constrained problem and hence let $\hat{x} \in \mathcal{F}$ be such that $f(\hat{x}) < f(x^*)$. By Assumption 23.1-(iii) and the continuity of f we can find another point $\tilde{y} \in H \cap D$

such that

$$f(\tilde{y}) < f(x^*). \quad (23.4)$$

On the other hand, as x_k is a minimizer of $\Psi(x; r_k)$ in $H \cap D$, we have

$$f(x_k) + r_k f_B(x_k) \leq f(\tilde{y}) + r_k f_B(\tilde{y})$$

and hence, for $k \in K, k \rightarrow \infty$ we have

$$f(x^*) + \liminf_{k \rightarrow \infty, k \in K} r_k f_B(x_k) \leq f(\tilde{y}),$$

so that, by (23.3), we have $f(x^*) \leq f(\tilde{y})$, which contradicts (23.4). This proves (ii). \square

Under convexity assumptions we can choose the barrier function and the merit function in a way that the minimization of Ψ is a convex problem. In fact, we can state the following proposition.

Proposition 23.2 (Convexity Conditions) *Suppose that*

- (a) *the set H is convex and $H \cap D \neq \emptyset$;*
- (b) *the functions f and g_i , for $i = 1, \dots, m$ are convex;*
- (c) *f_B is a barrier function and it is defined as $f_B(x) = I(g(x))$, where, for $g < 0$, the function $I : g(D) \rightarrow R$ is convex and isotone, in the sense that for every pair of vectors $u, v \in R^m$, such that $u < 0, v < 0$, and $u \leq v$ we have $I(u) \leq I(v)$.*

Then,

- (i) *the sets $H \cap D$ and \mathcal{F} are convex;*
- (ii) *the function f_B is convex on $H \cap D$;*
- (iii) *for every $r > 0$, the function $\Psi(x; r) = f(x) + rf_B(x)$, is a convex function of x on $H \cap D$.*

Proof Assertion (i) follows immediately from assumptions (a) and (b). We prove assertion (ii). Let $x, y \in H \cap D$. Then, by convexity of g we have, for $0 \leq \lambda \leq 1$,

$$g((1 - \lambda)x + \lambda y) \leq (1 - \lambda)g(x) + \lambda g(y) < 0,$$

and hence, as the mapping I is isotone, we can write

$$I(g((1-\lambda)x + \lambda y)) \leq I((1-\lambda)g(x) + \lambda g(y)).$$

By convexity of I , we have also

$$I((1-\lambda)g(x) + \lambda g(y)) \leq (1-\lambda)I(g(x)) + \lambda I(g(y)),$$

and therefore, from the two last inequalities, we obtain

$$I(g((1-\lambda)x + \lambda y)) \leq (1-\lambda)I(g(x)) + \lambda I(g(y)),$$

which proves (ii) and hence also (iii), because of the assumptions made. \square

It can be easily verified that the both the logarithmic barrier function and the inverse barrier function, already introduced, are convex and isotone functions defined on $g(D)$. In these cases the problem of minimizing Ψ on $H \cap D$ is a convex programming problem.

The main drawbacks of the barrier method described above are essentially two:

- the need of starting from a strict feasible point;
- the ill-conditioning of the potential function as the boundary of the feasible set is approached.

The first point may require, in general, some transformation of the original problem or the use of mixed interior and exterior methods. The second point constitutes a serious difficulty from a computational point of view and thus the interior methods were not considered very attractive for almost two decades, in comparison, for instance, with augmented Lagrangian methods.

In the late '70s started an intense research activity on the *computational complexity* of Linear Programming (LP). The first algorithm with polynomial complexity for LP, known as the *Ellipsoid method*, was proposed in 1979 by L. G. Khachiyan. This discovery had relevant theoretical consequences, but the algorithm was not computationally efficient, in comparison with the simplex method. In 1984 an interior point algorithm with polynomial complexity for LP was proposed by N. Karmarkar, who claimed a great computational efficiency in the solution of LP problems. The paper of Karmarkar has promoted significant advances in the development of new interior point algorithms, both for linear and nonlinear programming, which have greatly enhanced the relevance of these methods in the solution of convex problems.

Some basic ideas on interior point methods are outlined in the next sections.

23.2 Interior Point Methods for Linear Programming

23.2.1 Definitions and Notation

Let us consider the general linear programming (LP) problem in standard form, which we will assume as the *primal problem*:

$$\begin{aligned} & \min c^T x \\ & Ax = b \\ & x \geq 0, \end{aligned} \tag{23.5}$$

where $x \in R^n$, $c \in R^n$, $A : R^n \rightarrow R^m$ is a real $m \times n$ matrix and $b \in R^m$.

The *primal feasible set* is the set

$$\mathcal{F}_P = \{x \in R^n : Ax = b, x \geq 0\},$$

and the *primal strictly feasible set* (possibly empty) is defined by:

$$\mathcal{F}_P^0 = \{x \in R^n : Ax = b, x > 0\}.$$

To the primal problem we can associate the *dual problem*:

$$\begin{aligned} & \max b^T u \\ & A^T u \leq c, \end{aligned}$$

where $u \in R^m$. Using slack variables, the dual problem can be rewritten in the standard form

$$\begin{aligned} & \max b^T u \\ & A^T u + s = c \\ & s \geq 0, \end{aligned} \tag{23.6}$$

where $u \in R^m$ e $s \in R^n$. The *dual feasible set* is the set

$$\mathcal{F}_D = \{(u, s) \in R^m \times R^n : A^T u + s = c, s \geq 0\}$$

and the (possibly empty) *dual strictly feasible set* is

$$\mathcal{F}_D^0 = \{(u, s) \in R^m \times R^n : A^T u + s = c, s > 0\}.$$

We denote by Ω_P , Ω_D the sets of optimal primal and dual solutions respectively and we set

$$\Omega = \Omega_P \times \Omega_D.$$

23.2.2 A Summary of Basic LP Theory

From LP theory we recall the following results.

Proposition 23.3 (Weak Duality) *If x is a primal feasible point and u is dual feasible, we have $c^T x \geq b^T u$. \square*

Proposition 23.4 (Strong Duality and Optimality Conditions)

- (a) *Problem (23.5) has an optimal solution if and only if primal and dual feasible sets are non empty;*
- (b) *the point x^* is an optimal solution of Problem (23.5) if and only if there exists a pair (u^*, s^*) such that*

$$Ax^* = b, \quad x^* \geq 0, \quad A^T u^* + s^* = c, \quad s^* \geq 0, \quad c^T x^* = b^T u^*$$

and (u^, s^*) is an optimal solution of (23.6);*

- (c) *if the primal feasible set is non empty, then the primal objective function is bounded below on the primal feasible set if and only if the dual feasible set is non empty.* \square

The optimality conditions given at point (b) above are equivalent to the KKT conditions and can be reformulated by imposing complementarity conditions.

Proposition 23.5 (Complementarity) *The point x^* is an optimal solution of Problem (23.5) if and only if there exists (u^*, s^*) such that*

$$A^T u^* + s^* = c, \tag{23.7}$$

$$Ax^* = b, \tag{23.8}$$

$$(\text{complementarity}) \quad x_j^* s_j^* = 0, \quad j = 1, \dots, n, \tag{23.9}$$

$$(x^*, s^*) \geq 0. \tag{23.10}$$

The pair (u^, s^*) is an optimal solution of the dual problem.* \square

Let us define the diagonal matrices $X = \text{Diag}(x)$, $S = \text{Diag}(s)$ and denote by $e = (1, 1 \dots, 1)^T$ the unit vector in R^n . Then the complementarity condition (23.9)

can be written in the form

$$XSe = 0. \quad (23.11)$$

Thus, the solution of the LP problem can be obtained by finding a solution of the system

$$F(x, u, s) = \begin{pmatrix} A^T u + s - c \\ Ax - b \\ XSe \end{pmatrix} = 0, \quad (23.12)$$

that satisfies the nonnegativity conditions:

$$(x, s) \geq 0. \quad (23.13)$$

Problem (23.12) and (23.13) will be called *primal-dual problem*.

We indicate by \mathcal{F}_{PD} the *primal-dual feasible set*, that is, the set of primal and dual variables defined by:

$$\mathcal{F}_{PD} = \{(x, u, s) : A^T u + s = c, Ax = b, (x, s) \geq 0\},$$

and by \mathcal{F}_{PD}^0 the (possibly empty) *primal-dual strictly feasible set*, that is:

$$\mathcal{F}_{PD}^0 = \{(x, u, s) : A^T u + s = c, Ax = b, (x, s) > 0\}.$$

23.2.3 Main Classes of IPMs for LP and Basic Assumptions

Interior point methods for LP can be defined by specifying:

- the *space of variables* Y
- an open convex set $D \subseteq Y$ defined by linear inequalities, where the points generated by an IPM must remain;
- an affine subspace $H \subseteq Y$, defined by linear equality constraints;
- an *optimality measure* $\mu : D \rightarrow \mathbb{R}$, such that a “sufficient reduction” of μ allows us to approximate an optimal solution.

The optimality measure μ can depend on the (primal or dual) objective function or else it can measure the violation of the complementarity condition.

Variables and constraints may include primal and/or dual variables and constraints. More specifically, with reference to (23.5) we can distinguish:

- *primal methods*:

$$Y = \mathbb{R}^n, \quad D = \{x \in Y : x > 0\}, \quad H = \{x \in Y : Ax = b\};$$

- *dual methods:*

$$Y = R^m \times R^n, \quad D = \{(u, s) \in Y : s > 0\}, \quad H = \{(u, s) \in Y : A^T u + s = c\};$$

- *primal-dual methods:*

$$Y = R^n \times R^m \times R^n, \quad D = \{(x, u, s) \in Y : (x, s) > 0\},$$

$$H = \{(x, u, s) \in Y : A^T u + s = c, Ax = b\}.$$

An IPM can be defined under appropriate assumptions on the set where the algorithm must operate. Note that, in the general case, we may have problems where $H \cap D = \emptyset$, so that infeasible methods must be adopted.

When required, we will refer to the following assumptions.

Assumption 23.2 (Regularity Assumptions)

H1. *The strictly feasible primal set \mathcal{F}_P^0 is non empty, that is*

$$\mathcal{F}_P^0 = \{x \in R^n : Ax = b, x > 0\} \neq \emptyset.$$

H2. *The strictly feasible dual set \mathcal{F}_D^0 is non empty, that is:*

$$\mathcal{F}_D^0 = \{(u, s) \in R^m \times R^n : A^T u + s = c, s > 0\} \neq \emptyset.$$

□

If both assumptions H1 and H2 hold, it follows that also the strictly feasible primal-dual set is non empty, that is :

$$\mathcal{F}_{PD}^0 = \{(x, u, s) \in R^n \times R^m \times R^n : Ax = b, A^T u + s = c, (x, s) > 0\} \neq \emptyset.$$

We can state the following characterization of the preceding assumptions, which is proved in the appendix to this chapter.

Proposition 23.6 (Regularity and Optimality)

- (i) Suppose that the primal feasible set is non empty. Then the set of primal optimal solutions Ω_P is non-empty and bounded if and only if assumption H2 is satisfied.
- (ii) Suppose that the dual feasible set is non empty. Then the set of dual optimal solutions Ω_D is non-empty and bounded if and only if assumption H1 is satisfied.
- (iii) Suppose that the primal and dual feasible sets are non empty. Then the set of primal-dual optimal solutions $\Omega = \Omega_P \times \Omega_D$ is non-empty and bounded if and only if assumptions H1 and H2 are satisfied. \square

The most important distinction among IPMs methods for linear programming is that related to the solution strategy. We can distinguish, in particular:

- *affine scaling methods*
- *path-following methods*,
- *potential reduction methods*.

This distinction can be introduced, in principle, in each of the classes considered above, that is, in primal, dual or primal-dual methods.

Affine Scaling Methods

In these methods the primal optimality measure selected is minimized in a neighborhood B_k of the current point y_k belonging to D . The neighborhood is an ellipsoid (called *Dikin's ellipsoid*), whose structure is determined through an affine transformation dependent on y_k . The role of this transformation is that of scaling the variables, in a way that the new point y_{k+1} obtained from the minimization on B_k is not too close to the boundary of the feasible set. Thus, relatively large steps are permitted while remaining in D .

This technique, introduced by Dikin in 1967 [79] was rediscovered after the paper of Karmarkar, with the objective of simplifying his algorithm, by replacing the *projective* scaling considered by Karkarkar with a much simpler affine scaling technique. However the method in its original form, does not have the same theoretical properties of Karmarkar's method and the subsequent polynomial time modifications can be viewed as potential reduction methods.

Path-Following Methods

This class of methods consists in algorithms that follow approximately a path (called *central path*) in D , until an optimal solution is approximated with sufficient accuracy. The algorithms now considered to be more efficient for solving LP

problems are the *primal-dual path-following methods*, which will be described in the sequel. In this case the path-following method can also be viewed as an homotopy method that performs a sequence of Newton steps, for decreasing values of the parameter τ , for solving a modification of system (23.12), where the constraint $XSe = 0$ is replaced by $XSe = \tau e$.

Potential Reduction Methods

These methods are based on the use of a potential function Φ and consist in a sequence of steps that guarantee a sufficient reduction of Φ . Also in these techniques a transformation is performed for scaling the variables and the transformation can be projective, as in Karmarkar's method, or affine, as in affine scaling methods that do not make use of a potential function.

The best known potential functions used in IPMs make use of a logarithmic barrier function.

In particular, we can consider the following functions, defined on D .

Logarithmic Potential Functions

(a) primal logarithmic function:

$$\Phi(x; \tau) = c^T x - \tau \sum_{j=1}^n \log x_j;$$

(b) dual logarithmic function:

$$\Phi(u, s; \tau) = -b^T u - \tau \sum_{j=1}^n \log s_j;$$

(c) primal-dual logarithmic function

$$\Phi(x, s; \tau) = x^T s - \tau \sum_{j=1}^n \log(x_j s_j).$$

Actually, it can be shown that, in general, the distinction between the three classes of IPMs mentioned before is more of a quantitative than of a qualitative nature. In fact each IPM can be viewed as a sequence of steps in the space Y , such that the search direction is a suitable combination of two fundamental components d_a and d_{cent} . The component d_a has essentially the same role of the search direction in affine scaling methods and must realize a compromise between feasibility and optimality.

The component d_{cent} is a *centering* component that attempts to reach a *central path* in D that terminates in an optimal solution.

In the sequel first we will consider a class of feasible path-following methods (where $H \cap D \neq \emptyset$) and then we will illustrate the essential structure of an infeasible algorithm.

23.2.4 Feasible Path-Following Methods

We start by defining the notion of central path, which plays a fundamental role in these methods.

23.2.4.1 Central Path for Feasible Methods

Under the assumption that $H \cap D$ is non empty, the *central path* can be defined as the set of minimizers of a potential function for different values of the parameter τ that weights the barrier term. In particular, let us consider the primal logarithmic potential function and define the problem

$$\begin{aligned} \min \Phi(x; \tau) &= c^T x - \tau \sum_{j=1}^n \log x_j, \\ Ax &= b, \quad x > 0. \end{aligned} \tag{23.14}$$

We will establish existence and uniqueness of the solutions to this problem for each given $\tau > 0$. Preliminarily we state the following lemma.

Lemma 23.1 *Let $(\bar{x}, \bar{u}, \bar{s}) \in \mathcal{F}_{PD}$; then, for every $(x, u, s) \in \mathcal{F}_{PD}$ we have*

$$c^T x = \bar{u}^T b + \bar{s}^T x \tag{23.15}$$

$$b^T u = -\bar{x}^T s + c^T \bar{x} \tag{23.16}$$

$$x^T s = c^T x - b^T u = -c^T x + \bar{x}^T s - c^T \bar{x}. \tag{23.17}$$

Proof As the equality constraints are satisfied at points of \mathcal{F}_{PD} we have

$$c = A^T \bar{u} + \bar{s}, \quad b = A \bar{x}$$

and also $Ax = b$, $A^T u = c - s$. Then we can write

$$c^T x = (A^T \bar{u} + \bar{s})^T x = \bar{u}^T (Ax) + \bar{s}^T x = \bar{u}^T b + \bar{s}^T x;$$

$$b^T u = (A\bar{x})^T u = \bar{x}^T A^T u = \bar{x}^T (c - s) = -\bar{x}^T s + c^T \bar{x};$$

$$x^T s = x^T (c - A^T u) = c^T x - b^T u = c^T x + \bar{x}^T s - c^T \bar{x},$$

which establish the assertion. \square

Note, in particular, that if $(\bar{x}, \bar{u}, \bar{s})$ is fixed in \mathcal{F}_{PD}^0 , then Eq. (23.15) shows that the primal objective function can be expressed with an affine function such that the linear term has positive coefficients. Equation (23.17) shows that the quadratic term $x^T s$ is actually an affine function of (x, s) on the feasible set.

Now we can prove the following result.

Proposition 23.7 (Existence of a Minimizer of the Potential Function)

Suppose that assumptions H1, H2 hold. Then, for every $\tau > 0$ there exists a global minimum point of $\Phi(x; \tau)$ on $\mathcal{F}_P^0 = \{x : Ax = b, x > 0\}$.

Proof By the assumptions made, we can find a point $(\bar{u}, \bar{s}) \in \mathcal{F}_D^0$ and, by (23.15), we can write:

$$\begin{aligned} \Phi(x; \tau) &= \tau \left(\frac{1}{\tau} c^T x - \sum_{j=1}^n \log x_j \right) = \tau \sum_{j=1}^n \left(\frac{1}{\tau} \bar{s}_j x_j - \log x_j \right) + b^T \bar{u} \\ &\geq \tau \sum_{j=1}^n ((x_j \delta) - \log(x_j \delta) + \log \delta) + b^T \bar{u}, \end{aligned} \tag{23.18}$$

where

$$\delta = \frac{1}{\tau} \min_{j=1, \dots, n} \{\bar{s}_j\} > 0. \tag{23.19}$$

Letting $y_j = x_j \delta$ and defining the function

$$g(t) = t - \log t,$$

from (23.18) we obtain

$$\Phi(x; \tau) \geq \tau \sum_{j=1}^n g(y_j) + C, \quad (23.20)$$

where

$$C = \tau n \log \delta + b^T \bar{u}.$$

Now, for a given $\alpha \in R$, let us define the level set of Φ in \mathcal{F}_P^0 , that is

$$\mathcal{L}_\tau(\alpha) = \{x : Ax + b, x > 0, \Phi(x; \tau) \leq \alpha\}.$$

As \mathcal{F}_P^0 is non empty we can take, for instance, $\bar{x} \in \mathcal{F}_P^0$ and $\alpha = \Phi(\bar{x}; \tau)$, so that $\mathcal{L}_\tau(\alpha)$ is non empty. We show that every level set $\mathcal{L}_\tau(\alpha) \neq \emptyset$ is compact. First we prove that $\mathcal{L}_\tau(\alpha)$ is bounded. Reasoning by contradiction, suppose there exists a sequence $\{x_k\}$, with $x_k \in \mathcal{L}_\tau(\alpha)$, such that $\|x_k\| \rightarrow \infty$. In correspondence to these points, let us define $y_j^k = (x_k)_j \delta$, for $j = 1, \dots, n$, where δ is defined in (23.19). As $\|x_k\| \rightarrow \infty$, there must exist at least a j such that $y_j^k \rightarrow \infty$. By (23.20), noting that $g(t) \geq 0$ and $g(t) \rightarrow \infty$ for $t \rightarrow \infty$, we have that $\Phi(x_k; \tau) \rightarrow \infty$, which contradicts the assumption $x_k \in \mathcal{L}_\tau(\alpha)$. Then we can assert that the level set is bounded and we must prove that it is also closed.

As the level set is bounded, every sequence of points $\{x_k\}$ with $x_k \in \mathcal{L}_\tau(\alpha)$ has a limit point. Let $\{x_k\}$ be a sequence of these points converging to a point \bar{x} . As $x_k > 0$ and $Ax_k = b$ for all k , we have necessarily that $\bar{x} \geq 0$ and $A\bar{x} = b$. Therefore, if $\bar{x} \notin \mathcal{L}_\tau(\alpha)$, there must exist an index j such that $\bar{x}_j = 0$. By (23.20), this implies that $\Phi(x_k, \tau) \rightarrow \infty$, which contradicts the assumption $x_k \in \mathcal{L}_\tau(\alpha)$ for large values of k . Thus $\mathcal{L}_\tau(\alpha)$ is also closed and hence compact and there must exist a minimum point of Φ in \mathcal{F}_P^0 . \square

Proposition 23.7 guarantees that the problem of minimizing Φ on \mathcal{F}_P^0 has a solution and now we want to characterize this solution. We know that $\Phi(\cdot; \tau)$ is a strictly convex function on the convex set \mathcal{F}_P^0 . In fact we have

$$\nabla \Phi(x; \tau) = c - \tau X^{-1} e,$$

$$\nabla^2 \Phi(x; \tau) = \tau X^{-2},$$

where $X = \text{Diag}(x_j)$. As $x > 0$ and $\tau > 0$, the Hessian matrix is positive definite on \mathcal{F}_P^0 . Thus we can state the following proposition.

Proposition 23.8 (Characterization of the Central Path) Suppose that assumptions H1, H2 hold. Then, for every $\tau > 0$ there exists a unique optimal solution of Problem (23.14). Moreover, a point $x(\tau)$ is an optimal solution if and only if there exist $(u(\tau), s(\tau)) \in R^m \times R^n$ such that

$$A^T u(\tau) + s(\tau) = c, \quad (23.21)$$

$$Ax(\tau) = b, \quad (23.22)$$

$$X(\tau)S(\tau)e = \tau e, \quad (23.23)$$

$$(x(\tau), s(\tau)) > 0. \quad (23.24)$$

Proof The existence of an optimal solution follows from Proposition 23.7 and uniqueness is a consequence of the strict convexity of Φ . As the constraints are linear, the KKT conditions are necessary and sufficient conditions for optimality. Since $(x, s) > 0$, we can consider only equality constraints. Thus $x(\tau)$ is the optimal solution if and only if $Ax(\tau) = b$ (and hence (23.22) holds) and, moreover, there exist Lagrange multipliers $\lambda \in R^m$ such that

$$c - \tau X(\tau)^{-1}e + A^T \lambda = 0. \quad (23.25)$$

Letting $u(\tau) = -\lambda$, from (23.25) we have

$$A^T u(\tau) + \tau X(\tau)^{-1}e = c.$$

Then, assuming $s(\tau) = \tau X(\tau)^{-1}e > 0$ inequality (23.24) holds and we obtain (23.21). We have also $X(\tau)s(\tau) = \tau e$, which can be rewritten in the form $X(\tau)S(\tau)e = \tau e$, and equality (23.23) holds. Conversely, if the conditions stated hold, the KKT conditions are satisfied by taking $\lambda = -u(\tau)$. \square

The preceding proposition guarantees that, for a fixed $\tau > 0$, system (23.21)–(23.24) has a unique solution. We note that for $\tau = 0$ we obtain the optimality condition of LP. The solutions of (23.21)–(23.24) define in Y a trajectory in the interior of D , which constitutes the *central path*. Thus, the sets $\mathcal{C}_P = \{x(\tau) : \tau > 0\}$, $\mathcal{C}_D = \{(u(\tau), s(\tau)) : \tau > 0\}$, $\mathcal{C}_{PD} = \{(x(\tau), u(\tau), s(\tau)) : \tau > 0\}$ define, respectively, the primal, dual or primal-dual central path. For $\tau \rightarrow \infty$ the primal central path \mathcal{C}_P terminates in the unique solution of the problem

$$\begin{aligned} \min f_B(x) &= - \sum_{j=1}^n \log x_j, \\ Ax &= b, \quad x > 0, \end{aligned} \quad (23.26)$$

and this point is called the *analytic center* of \mathcal{F}_P .

Remark 23.1 In general, the analytic center [241] of a given polyhedron, represented as a system of inequalities

$$a_i^T x \leq b_i, \quad i = 1, \dots, m,$$

is the solution of the problem

$$\min - \sum_{j=1}^n \log(b_j - a_j^T x),$$

defined for x such that $b_i - a_i^T x > 0$ for all i .

Assuming that the constraints are normalized, so that $\|a_i\| = 1$, for all i , the analytic center can be interpreted as the point that maximizes the product of the distances $b_i - a_i^T x$, $i = 1, \dots, m$ from the hyperplanes that define the polyhedron. It depends on the mathematical representation of the system and changes with the addition of other redundant strict inequalities. \square

23.2.4.2 Primal-Dual Path-Following Methods

We restrict our attention to primal-dual path-following methods that generate a sequence of points (x_k, u_k, s_k) in $D_{PD} = \{(x, u, s) : (x, s) > 0\}$, starting from an initial point $(x_0, u_0, s_0) \in D_{PD}$. These methods approximate the central path C_{PD} by approximating a solution of the system of nonlinear equations

$$F_\tau(x, u, s) = \begin{pmatrix} A^T u + s - c \\ Ax - b \\ XSe - \tau e \end{pmatrix} = 0. \quad (23.27)$$

with

$$(x, s) > 0.$$

The approximation to the central path can be based on a sequence of iterations of the form

$$(\bar{x}, \bar{u}, \bar{s}) = (x, u, s) + \alpha(dx, du, ds),$$

where $\alpha \in R$ is the step-size and $(dx, du, ds) = d_n$ is the Newton's direction, obtained by solving the system $J(x, u, s)d_n = -F_\tau(x, u, s)$, where $J(x, u, s)$ is

the Jacobian matrix of F_τ . We have

$$J(x, u, s) = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix},$$

and hence the Newton's direction can be computed by solving the system:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ du \\ ds \end{pmatrix} = - \begin{pmatrix} A^T u + s - c \\ Ax - b \\ XSe - \tau e \end{pmatrix}. \quad (23.28)$$

Thus, feasible methods generate points that belong to the set

$$\mathcal{F}_{PD}^0 = \{(x, u, s) : A^T u + s = c, Ax = b, (x, s) > 0\},$$

starting from $(x_0, u_0, s_0) \in \mathcal{F}_{PD}^0$. As the current point (x, u, s) satisfies the equality constraints, Newton's direction could be computed by solving the system:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ du \\ ds \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ XSe - \tau e \end{pmatrix}. \quad (23.29)$$

The solution of this system yields a search direction such that

$$Adx = 0, \quad A^T du + ds = 0,$$

so that the updated point $(\bar{x}, \bar{u}, \bar{s})$ still satisfies the equality constraints.

We assume that Newton's direction is computed by solving system (23.29). In addition to H1,H2 we also assume that the following condition holds.

Assumption 23.3

H3. *The matrix A has rank m .*

□

In order to compute a solution to system (23.29) we set

$$r = XSe - \tau e \quad (23.30)$$

and we rewrite the system in the form

$$A^T du + ds = 0, \quad (23.31)$$

$$Adx = 0, \quad (23.32)$$

$$Sdx + Xds = -r. \quad (23.33)$$

From (23.33) we get

$$ds = -X^{-1}Sdx - X^{-1}r,$$

and substituting ds in (23.31) we get a system in the components dx, du :

$$A^T du - X^{-1}Sdx = X^{-1}r, \quad (23.34)$$

$$Adx = 0, \quad (23.35)$$

By solving the first equation with respect to dx we have

$$dx = -S^{-1}r + S^{-1}XA^T du,$$

and hence, by (23.32) we obtain

$$AS^{-1}XA^T du = AS^{-1}r, \quad (23.36)$$

where the matrix $AS^{-1}XA^T$ is non singular by assumption H3. Letting

$$D = S^{-1/2}X^{1/2},$$

Eq. (23.36) becomes

$$AD^2A^T du = AS^{-1}r$$

and hence we can write

$$du = (AD^2A^T)^{-1}AS^{-1}r, \quad (23.37)$$

$$ds = -A^T du = -A^T(AD^2A^T)^{-1}AS^{-1}r, \quad (23.38)$$

$$dx = -S^{-1}r - S^{-1}Xds = -S^{-1}r + D^2A^T(AD^2A^T)^{-1}AS^{-1}r. \quad (23.39)$$

The above expressions (not to be used in computation!) show that system (23.29) admits a unique solution for each fixed value of τ .

The various primal-dual path-following methods differ essentially in the criterion used for defining and updating τ and in the rule followed for the choice of the step-size α . Usually the optimality criterion adopted is a measure of the violation of the

complementarity condition, of the form

$$\mu = \frac{x^T s}{n},$$

and τ is defined by

$$\tau = \sigma \mu,$$

where $\sigma \in [0, 1]$ is called *centering parameter*. It can be easily verified that for $\sigma = 0$ (and hence $\tau = 0$) system (23.29) defines the Newton's direction related to system (23.12), while for $\sigma = 1$ we have a Newton's step towards the central path. For $\sigma_k \rightarrow 0$, the point of the central path that we are approximating converges towards the optimal solution.

The dependence of the updated optimality measure $\bar{\mu}$ on σ and α is established in the next proposition.

Proposition 23.9 Consider the iteration defined by

$$(\bar{x}, \bar{u}, \bar{s}) = (x, u, s) + \alpha(dx, du, ds),$$

where the search direction is the solution of system (23.29) with $\tau = \sigma \mu$.

Then we have

$$dx^T ds = 0, \quad (23.40)$$

and the updated value of the optimality criterion is given by:

$$\bar{\mu} = (1 - \alpha(1 - \sigma)) \mu. \quad (23.41)$$

Proof From system (23.29), taking the scalar product of dx and the first equation we have:

$$dx^T A^T du + dx^T ds = (Adx)^T du + dx^T ds = 0,$$

so that by the second equation $Adx = 0$, we obtain (23.40). From the third equation, letting $\tau = \sigma \mu$ we obtain

$$Sdx + Xds = -(XSe - \sigma \mu e),$$

which can be written, in terms of the single components, as

$$s_j dx_j + x_j ds_j = -(x_j s_j - \sigma \mu), \quad j = 1, \dots, n.$$

Therefore, summing both members of these equations we get:

$$s^T dx + x^T ds = -(x^T s - n\sigma \mu) = -(1 - \sigma)x^T s. \quad (23.42)$$

On the other hand we have:

$$\bar{x}^T \bar{s} = (x + \alpha dx)^T (s + \alpha ds) = x^T s + \alpha(s^T dx + x^T ds) + \alpha^2 dx^T ds,$$

and hence, taking into account (23.40) and (23.42), we obtain (23.41). \square

The choice of σ and of the step-size α must guarantee that the constraints $(x, s) > 0$ are still satisfied at the updated point and, at same time, that the step-size is large enough, to have a “sufficient” reduction of μ .

In path-following methods this is obtained by defining at each step a suitable neighborhood \mathcal{N} of the central path where the updated point has to be placed. We must require that:

- the central path C_{PD} is contained in \mathcal{N}
- \mathcal{N} is contained in \mathcal{F}_{PD}^0 .

On the basis of the size of this neighborhood we can distinguish, in particular:

- *short step* path-following (SPF) methods;
- *long step* path-following (LPF) methods.

In SPF the neighborhood of the central path is of the form

$$\mathcal{N}_2(\theta) = \{(x, u, s) \in \mathcal{F}_{PD}^0 : \|XSe - \mu e\|_2 \leq \theta \mu\},$$

where $\|\cdot\|_2$ is the Euclidean norm and $\theta \in (0, 1)$ (typically $\theta = 0.5$).

In LPF we consider a much larger neighborhood, as

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, u, s) \in \mathcal{F}_{PD}^0 : x_j s_j \geq \gamma \mu, \quad j = 1, \dots, n\},$$

with $\gamma \approx 10^{-3}$.

By imposing that

$$(x, u, s) \in \mathcal{N}_{-\infty}(\gamma)$$

we essentially require that the single products $x_j s_j$ are reduced not much faster than the mean

$$\mu = \frac{s^T x}{n},$$

in order to prevent the possibility that the algorithm reaches the boundary of \mathcal{P}_{PD}^0 when we are still far from the optimal solution on the central path. On the other hand, large values of α yield a greater reduction of μ as shown in Proposition 23.9. A conceptual model of a LPF algorithm, taken from [260], where a compromise between these requirements is reached, is outlined below.

LPF Algorithm Model

- 0. Data** $\gamma \in (0, 1)$, $0 < \sigma_{\min} < \sigma_{\max} < 1$

- 1. (Initialization)**

Determine a point $(x_0, u_0, s_0) \in \mathcal{N}_{-\infty}(\gamma)$ and set $k = 0$.

- 2. For** $k = 0, 1, 2, \dots$

Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$

Set $\mu_k = \frac{x_k^T s_k}{n}$, and solve the system

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ du \\ ds \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ X_k S_k e - \sigma_k \mu_k e \end{pmatrix}, \quad (23.43)$$

to obtain the direction (dx_k, du_k, ds_k) .

Compute the largest value of $\alpha \in [0, 1]$ such that

$$(x_{k+1}, u_{k+1}, s_{k+1}) = (x_k, u_k, s_k) + \alpha_k(dx_k, du_k, ds_k) \in \mathcal{N}_{-\infty}(\gamma).$$

End for

It can be shown that the algorithm is well defined and, in particular, that we can compute a value of α_k such that $(x_{k+1}, u_{k+1}, s_{k+1}) \in \mathcal{N}_{-\infty}(\gamma)$ and that, at the same time, we obtain a sufficient reduction of μ .

The following proposition can be proved [260].

Proposition 23.10 *Let $\varepsilon > 0$ and $\gamma \in (0, 1)$; suppose that $(x_0, u_0, s_0) \in \mathcal{N}_{-\infty}(\gamma)$ and that assumptions H1, H2, H3 are satisfied. Then, if the starting point satisfies $\mu_0 \leq 1/(\varepsilon^\delta)$, for some $\delta > 0$, there exists an index K , with*

$$K = O(n|\log \varepsilon|),$$

such that $\mu_k \leq \varepsilon$, for all $k \geq K$. □

It has been shown that path-following methods can be implemented in a way that a polynomial time complexity in the Turing model can be achieved, for LP problems with integer coefficients.

Actually, path-following methods cannot determine for finite values of k an optimal solution, since $\mu_k > 0$ for all k and hence the complementarity condition cannot be satisfied. However, a *termination procedure* can be implemented such that a vertex solution can be computed starting from a good approximation of an optimal solution. In particular, because of the fact that the coefficients are integer, it has been shown that this is possible when the optimality measure μ_k is inferior to a threshold value, that is when

$$\mu_k = x_k^T s_k / n \leq e^{-tL}, \quad t = 10 \log_e 2,$$

where L is the total length of the data in a logarithmic model. Thus Proposition 23.10 guarantees, under the assumptions stated, that the LPF method yields a polynomial time algorithm for LP.

23.2.5 Infeasible Methods

The feasible methods described above require starting from a strict feasible point $(x_0, u_0, s_0) \in \mathcal{F}_{PD}^0$. In many cases, however, this set can be empty and we should reformulate the problem. An alternative, more convenient, approach can be that of admitting that feasibility of the equality constraints is reached only in the limit of an iterative process, while positivity constraints on (x, s) are always satisfied. In this case, given $(x, s) > 0$ we can define the *residuals*

$$r_b = Ax - b, \quad r_c = A^T u + s - c,$$

and then compute the Newton's direction by means of a system of the form

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ du \\ ds \end{pmatrix} = - \begin{pmatrix} r_b \\ r_c \\ XSe - \sigma\mu e \end{pmatrix}. \quad (23.44)$$

We can describe the essential features of a primal-dual algorithm, similar to algorithm LPF, where it is constructed appropriately a neighborhood that includes also infeasible points.

More specifically, we can define the set

$$\begin{aligned} \mathcal{N}_{-\infty}(\gamma, \beta) = \{(x, u, s) : \|r_b, r_c\| \\ \leq (\|r_b^0, r_c^0\|/\mu_0) \beta \mu, x_j s_j \geq \gamma \mu, j = 1, \dots, n\}, \end{aligned}$$

where $\gamma \in (0, 1)$, $\beta \geq 1$, and r_b^0, r_c^0, μ_0 are evaluated at a starting point $(x_0, u_0, s_0) \in \mathcal{N}_{-\infty}(\gamma, \beta)$.

In this way, if μ is forced to zero we have also that r_b, r_c are forced to zero. This requires that the step-size α along the search direction satisfies also an Armijo-type condition on μ .

The following algorithm, introduced in [260]) and called IPF (Infeasible Path-Following), can be defined.

IPF Algorithm Model

0. Data $\gamma \in (0, 1)$, $\beta \geq 1$, $0 < \sigma_{\min} < \sigma_{\max} < 0.5$

1. (Initialization)

Choose a point $(x_0, s_0) > 0$ and set $k = 0$.

2. For $k = 0, 1, 2, \dots$

Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$

Set $\mu_k = \frac{x_k^T s_k}{n}$, and solve the system

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S_k & 0 & X_k \end{pmatrix} \begin{pmatrix} dx_k \\ du_k \\ ds_k \end{pmatrix} = - \begin{pmatrix} r_b^k \\ r_c^k \\ X_k S_k e - \sigma_k \mu_k e \end{pmatrix}, \quad (23.45)$$

to obtain the direction (dx_k, du_k, ds_k) .

Compute the largest value α_k of $\alpha \in [0, 1]$ such that

$$\mu_{k+1} \leq (1 - 0.01\alpha_k)\mu_k$$

and

$$(x_{k+1}, u_{k+1}, s_{k+1}) = (x_k, u_k, s_k) + \alpha_k(dx_k, du_k, ds_k) \in \mathcal{N}_{-\infty}(\gamma, \beta).$$

End for

It can be shown that the algorithm is well defined and that the sequence $\{\mu_k\}$ converges to zero. The proof is much more complicated than in the feasible case. We report the conclusion established in [260].

Proposition 23.11 *The sequence $\{\mu_k\}$ generated by Algorithm IPM converges Q -linearly to zero and the sequence of the residuals $\{\|(r_b^k, r_c^k)\|\}$ converges R -linearly to zero.* \square

Under appropriate conditions on the starting point it can also be established the polynomial complexity of the algorithm by proving an analogous of Proposition 23.10.

23.3 Extensions

IPMs have had a major role in the development of convex optimization, by providing efficient computational algorithms, based on Newton's method, which retain, under mild assumptions, a polynomial-time complexity. In particular, primal-dual IPMs have been extended to quadratic programming problems, linear complementarity, semidefinite programming, cone programming. Also extensions to non convex nonlinear programming problems have been studied and experimented.

Here we will confine ourselves to sketch the extension of primal-dual path-following methods to quadratic programming and to nonconvex problems in the special case where there exist strictly positive feasible points. For further information and for the description of infeasible algorithms the reader is addressed to the literature mentioned in the notes.

23.3.1 Quadratic Programming

IPMs can be extended to convex quadratic programming problems of the form:

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{QP}$$

where $x \in R^n$, $c \in R^n$, $A : R^n \rightarrow R^m$, $b \in R^m$, Q is a symmetric positive definite $n \times n$ matrix. In this case, using the KKT conditions, the necessary and sufficient optimality conditions can be put into the form:

$$\begin{aligned} Ax &= b \\ A^T u + s - Qx &= c \\ XSe &= 0 \\ (x, s) &\geq 0. \end{aligned} \tag{23.46}$$

In a way similar to that followed in LP we can add to the objective function a logarithmic barrier term, and we can consider the problem

$$\begin{aligned} \min \Phi(x; \tau) &= \frac{1}{2} x^T Q x + c^T x - \tau \sum_{j=1}^n \log x_j, \\ Ax &= b, \quad x > 0. \end{aligned} \tag{23.47}$$

Under the assumption that primal and dual strictly feasible sets are non empty, it can be shown that we can construct a primal-dual central path as solution of the following system, for different values of τ :

$$\begin{aligned} Ax &= b \\ A^T u + s - Qx &= c \\ XSe &= \tau e \\ (x, s) &> 0. \end{aligned} \tag{23.48}$$

Then we can define path-following algorithms based on Newton's method and we can achieve polynomial complexity results.

23.3.2 IPMs for Nonconvex Problems

In nonconvex problems we can extend path-following methods, starting from KKT conditions. Without loss of generality, we can consider an inequality constrained problem of the form

$$\begin{aligned} \min f(x) \\ g(x) \leq 0, \end{aligned}$$

where $g : R^n \rightarrow R^m$ and we assume that the problem functions are twice continuously differentiable.

By introducing a vector of nonnegative slack variables $s \in R^m$ we obtain the equivalent problem

$$\begin{aligned} \min f(x) \\ g(x) + s = 0, \\ s \geq 0. \end{aligned}$$

Then the KKT conditions, with simple manipulations, can be written in the form:

$$\begin{aligned}\nabla f(x) + \nabla g(x)z &= 0 \\ g(x) + s &= 0 \\ Sz &= 0 \\ (z, s) &\geq 0,\end{aligned}\tag{23.49}$$

where $z \in R^m$ and $S = \text{Diag}(s)$. Under the assumption of strict feasibility, we can construct again a barrier problem by imposing that the vector s remains positive, so that we obtain a problem of the form:

$$\begin{aligned}\min_{x,s} \quad & f(x) - \tau \sum_{i=1}^m \log s_i \\ \text{subject to} \quad & g(x) + s = 0,\end{aligned}$$

where $\tau > 0$.

In this case, as we suppose that s remains in the open set $\{y \in R^m : y > 0\}$, we can apply the Lagrange multiplier rule, by ignoring the constraint $s > 0$.

Then the optimality conditions become

$$\begin{aligned}\nabla f(x) + \nabla g(x)z &= 0 \\ Sz - \tau e &= 0 \\ g(x) + s &= 0.\end{aligned}\tag{23.50}$$

In order to solve the non linear system at the point (x, s, z) we can compute the Newton's direction (dx, ds, dz) by solving the system

$$\begin{pmatrix} \nabla^2 L_{xx}(x, s) & 0 & \nabla g(x) \\ 0 & Z & S \\ \nabla g(x)^T & I & 0 \end{pmatrix} \begin{pmatrix} dx \\ ds \\ dz \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + \nabla g(x)z \\ Sz - \tau e \\ g(x) + s \end{pmatrix}. \tag{23.51}$$

where $L(x, s, z)$ is the Lagrangian function of the barrier problem and

$$\nabla^2 L_{xx}(x, s) = \nabla^2 f(x) + \sum_{i=1}^m z_i \nabla^2 g_i(x).$$

Using this iteration or some modified iteration, we can construct path-following algorithms based on line searches or on trust region techniques. The general idea is that of reducing the parameter τ , in a way that an approximation of a solution to (23.50) can converge (under appropriate assumptions) to a solution of (23.49), which yields a point satisfying the KKT conditions.

23.4 Appendix: Regularity and Optimality

We establish the following propositions, by employing the theorems of the alternative (see the appendix to Chap. 7).

Proposition 23.12 *Suppose that the primal feasible set is non empty, that is*

$$\mathcal{F}_P = \{x \in R^n : Ax = b, x \geq 0\} \neq \emptyset.$$

Then \mathcal{F}_P is bounded if and only if there exists a pair (\bar{u}, \bar{s}) such that

$$A^T \bar{u} + \bar{s} = 0, \bar{s} > 0.$$

Proof The set \mathcal{F}_P is bounded if and only if the system

$$Ay = 0, \quad y \geq 0, \quad e^T y > 0$$

has no solution and then, by Gale theorem of the alternative, if and only if there exists \bar{u} such that $A^T \bar{u} \leq -e < 0$ and hence if and only if there exist \bar{u}, \bar{s} such that

$$A^T \bar{u} + \bar{s} = 0, \quad \bar{s} > 0.$$

□

An immediate consequence of the preceding proposition is the following.

Proposition 23.13 *Suppose that $\mathcal{F}_P \neq \emptyset$. Then \mathcal{F}_P is bounded if and only if \mathcal{F}_D^0 is unbounded.*

Proof Suppose first that \mathcal{F}_P is non empty and bounded. Then the primal problem has an optimal solution (by Weierstrass theorem) and hence the dual feasible set is non empty, that is, there exist (\tilde{u}, \tilde{s}) such that $A^T \tilde{u} + \tilde{s} = c, \tilde{s} \geq 0$. By Proposition 23.12, there exist \bar{u}, \bar{s} such that $A^T \bar{u} + \bar{s} = 0, \bar{s} > 0$ and therefore, for every $\theta > 0$, letting $s(\theta) = \tilde{s} + \theta \bar{s} > 0$ and $u(\theta) = \tilde{u} + \theta \bar{u}$ we have $(u(\theta), s(\theta)) \in \mathcal{F}_D^0$. This implies that \mathcal{F}_D^0 is non empty and unbounded. Conversely, assume that $\mathcal{F}_P \neq \emptyset$ and that \mathcal{F}_D^0 is non empty and unbounded. This implies that there must exist \bar{u}, \bar{s} such that $A^T \bar{u} + \bar{s} = 0, \bar{s} > 0$; then, by Proposition 23.12 we have that \mathcal{F}_P is bounded. □

Let now $\alpha \in R$ and suppose that the primal level set $\mathcal{L}_P(\alpha)$ is non empty, that is

$$\mathcal{L}_P(\alpha) = \{x \in R^n : Ax = b, x \geq 0, c^T x \leq \alpha\} \neq \emptyset.$$

It is easily seen that if \mathcal{F}_P is bounded, also $\mathcal{L}_P(\alpha)$ is bounded. Then we state the following result.

Proposition 23.14 *Let $\alpha \in R$ and suppose that the primal level set $\mathcal{L}_P(\alpha)$ is non empty, that is*

$$\mathcal{L}_P(\alpha) = \{x \in R^n : Ax = b, x \geq 0, c^T x \leq \alpha\} \neq \emptyset,$$

where $c \neq 0$. Then, $\mathcal{L}_P(\alpha)$ is bounded if and only if the dual strictly feasible set \mathcal{F}_D^0 is non empty.

Proof We observe preliminarily that $\mathcal{L}_P(\alpha) \neq \emptyset$, implies also that $\mathcal{F}_P \neq \emptyset$. Suppose first that $\mathcal{L}_P(\alpha)$ is bounded. Then the homogeneous system

$$Ay = 0, \quad y \geq 0, \quad c^T y \leq 0, \quad e^T y > 0 \quad (23.52)$$

has no solution and hence, by Motzkin theorem of the alternative, there exist $z, w \geq 0, \lambda > 0, \sigma \geq 0$ such that

$$(\lambda e + w) + A^T z = \sigma c, \quad (\lambda e + w) > 0. \quad (23.53)$$

Letting $\tilde{s} = \lambda e + w > 0$, we have that if system (23.53) has solution there exist a solution z, \tilde{s}, σ to the system

$$A^T z + \tilde{s} = \sigma c, \quad \sigma \geq 0, \quad \tilde{s} > 0. \quad (23.54)$$

Now, if $\sigma = 0$ Proposition 23.12 implies that \mathcal{F}_P is bounded and hence Proposition 23.13 in turn implies that \mathcal{F}_D^0 is non empty. If $\sigma > 0$ letting $\hat{z} = z/\sigma$ and $\hat{s} = \tilde{s}/\sigma$, we can write, by (23.54)

$$A^T \hat{z} + \hat{s} = c, \quad \hat{s} > 0 \quad (23.55)$$

and this implies that \mathcal{F}_D^0 is non empty.

Suppose now that \mathcal{F}_D^0 is non empty. It is easily seen that this is equivalent to say that $-A^T z > -c$ has a solution and hence to require that the system

$$\begin{pmatrix} -A^T & c \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} z \\ \xi \end{pmatrix} \geq \begin{pmatrix} e \\ 1 \end{pmatrix}$$

has a solution. By Gale theorem of the alternative this implies that there exist y, ρ such that the system

$$Ay = 0, c^T y + \rho = 0, \quad e^T y + \rho > 0, \quad y \geq 0, \quad \rho \geq 0,$$

has no solution. Noting that $e^T y = 0$ would imply $y = 0$ and hence $\rho = 0$, which would contradict $e^T y + \rho > 0$, this is equivalent to say that the homogeneous system

$$Ay = 0, \quad y \geq 0, \quad c^T y \leq 0, \quad e^T y > 0$$

has no solution and hence that $\mathcal{L}_P(\alpha)$ is bounded, \square

Now consider the set Ω_P of the primal optimal solutions. We prove the following proposition.

Proposition 23.15 Suppose that $\mathcal{F}_P \neq \emptyset$ and that $c \neq 0$. Then the set of primal optimal solutions Ω_P is non-empty and bounded if and only if \mathcal{F}_D^0 is non empty.

Proof Suppose first that Ω_P is non-empty and bounded and let α^* be the optimal value of the objective function. This obviously implies that $\mathcal{L}_P(\alpha^*)$ is non empty and bounded and that \mathcal{F}_P is non-empty. Thus, the assertion follows from Proposition 23.14.

Conversely, suppose that \mathcal{F}_D^0 is non empty. As \mathcal{F}_P is non empty by assumption, we have that the set $\mathcal{L}_P(\alpha)$ is non empty for some α . Then, by Proposition 23.14, we have that $\mathcal{L}_P(\alpha)$ is bounded and hence compact. This obviously implies that Ω_P is non empty and bounded. \square

Recalling that the dual of the dual problem is the primal problem we can reformulate the preceding results by interchanging the role of the primal and the dual.

23.5 Notes and References

The book of Fiacco and McCormick [92], already mentioned in Chap. 21 has been the first fundamental contribution also to the study of interior methods and it contains references to early papers on barrier methods. The basic work on computational complexity in the late 70's has been the book [105]. The polynomial time- complexity of LP was established in [155] and the paper of Karmarkar that promoted the resurgence of interests on interior method was [150]. In almost twenty years from Karmarkar's paper the field of interior point methods has greatly expanded and several thousand papers on this subject has been published. Here we

can only mention some references on which we have based our introduction and some advanced works. In particular, we mention the books [260], [31], and [192]. Recent works are devoted to “modern” convex optimization (see, for instance, [14]), to the improvement of the linear algebra employed in interior point methods [62] and to the extension of IPM to non convex problems [196].

Chapter 24

Nonmonotone Methods



In this chapter we introduce some globalization techniques for solving minimization problems and nonlinear equations, which relax the descent requirements usually imposed on the objective function. We first discuss the basic motivation of this class of methods and then we describe non monotone line searches and non monotone globalization schemes.

24.1 Motivation and Basic Concepts

The algorithms studied in the preceding chapters are typically represented as *monotone methods*, that is as sequences of points, where the objective function (or some merit function, in constrained problems) is monotonically reduced (or, at least, non increased) at each major step. This feature has an obvious motivation: as we are searching for minimum points of a function, we may prefer to perform a sequence of *descent steps*. In most of cases, this also ensures that we remain in a level set corresponding to the function value at the starting point. Thus, if the level set is compact and the starting point is not a critical point, we can guarantee that the sequence generated by the algorithm has limit points and that these points at least improve our objective with respect to the starting value. If every subsequence of points converging to a limit point admits a subsequence with strictly decreasing function values, it is also possible to exclude that we are reaching asymptotically a limit point which is a local maximizer. However, these useful features are paid with some possible inefficiencies that can be quite relevant in some cases.

Let us first consider descent methods in the unconstrained case. We refer in the sequel to algorithms based on linesearches, but similar problems may arise in case of globalization strategies for trust region methods.

The main limitations of monotone methods is that we must often reduce the step-sizes along a search direction, in order to ensure monotonicity, but this reduction may have catastrophic effects in some situations.

A first case is when the step-size should have some prescribed value for obtaining a good convergence speed. The typical case is that of Newton's method, where ideally we would like to use a unit step-size at least in the convergence region of the pure Newton iteration. We have seen that globalization strategies can be implemented in a way that, under appropriate assumptions, the unit step-size can be accepted in a neighborhood of a solution. However, computational experience shows that the adoption of the full Newton step is often beneficial even in the early stages of the minimization process, in spite of possible temporary increases in the objective function values. In fact, we cannot easily determine, in practice, when we are entering the convergence region of Newton's method.¹ As we will see in Chap. 25, there are also recent implementations of the gradient method, like the *Barzilai-Borwein gradient method* [11], which are based on the adoption of a suitable step-size at each step, which may not correspond to an immediate reduction of the objective function.

Another difficult case for all monotone methods can be that of minimizing highly nonlinear functions in the presence of narrow curved valleys, since the attempt of reducing the objective function may cause the algorithm to be trapped at the bottom of the valley. In the extreme cases, this corresponds to a *linesearch failure*, which causes premature termination of the algorithm.

Monotone methods can also incur in the presence of many irrelevant local minimizers, due, for instance, to the presence of noise in the objective function and this may cause termination at one of these points.

Similar difficulties may arise in the context of constrained minimization methods. We have already discussed in Chap. 22 the so called *Maratos effect* in the globalization of Newton-type sequential quadratic programming methods, through the adoption of *non differentiable exact penalty functions*. In this case (because of non differentiability) the linesearch may not accept the unit step-size even if the algorithm is converging to a limit point that satisfies the assumptions which guarantee a superlinear convergence rate with a unit step-size. Monotone methods can also suffer from the ill-conditioning of the Hessian matrix, due to large values of penalty parameters in the minimization of penalty and augmented Lagrangian methods.

On the basis of the preceding observations, we are interested in methods that retain the most important features of monotone methods, but relax to some extent the monotonicity requirement.

Actually, non monotone methods have been considered in the literature since many years and many connection can be established with Lyapunov criteria for

¹ It was experienced that in a large set of test problems on the solution of nonlinear equations the pure Newton iterations converged (not monotonically with reference to an error measure) without any globalization control [136].

the stability analysis of discrete dynamical systems. Because of space limitations, here we will concentrate our exposition on some nonmonotone techniques for unconstrained problems, based on a combination of the non monotone linesearch proposed in [122] with a *watchdog technique* [46], along similar lines to that followed in [125]. More specifically, we will refer to the technique defined in [134] and extended also to derivative free methods, which has been largely experimented by the authors in many computational applications.

References to (some of the) other nonmonotone strategies proposed in the literature and to the application to trust region problems will be given in the notes at the end of this chapters

In the sequel we first introduce the non monotone *reference value* used in our acceptability criterion and we establish some preliminary result. Then we define various non monotone linesearches by extending the corresponding monotone schemes; we introduce our definition of watchdog rules and we prove the convergence of a *nonmonotone watchdog algorithm* combined with nonmonotone linesearches. We use,in particular, algorithms with this structure, in nonmonotone globalization schemes for Newton-type methods, both in unconstrained minimization and in the solution of nonlinear equations.

24.2 Convergence Issues in Nonmonotone Methods

We briefly discuss in an informal way the main technical issues underlying the convergence theory of nonmonotone methods. To this aim, first we recall the key points to ensure the global convergence of a monotone line search method. As already seen in preceding chapters, in a monotone line search-based framework, for each k we have that:

- a descent direction d_k is available;
- a stepsize α_k along d_k is computed by a line search satisfying the condition of sufficient reduction

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma \alpha_k \nabla f(x_k)^T d_k. \quad (24.1)$$

Therefore, as $f(x_{k+1}) \leq f(x_k)$, by assuming that f is bounded below, we have that the sequence of function values $\{f(x_k)\}$ converges, so that, the properties of the line search ensure that

$$\nabla f(x_k)^T d_k \rightarrow 0.$$

This latter condition, together with an *angle condition* on d_k , implies that

$$\nabla f(x_k) \rightarrow 0.$$

Then, the convergence of the sequence of function values $\{f(x_k)\}$ is fundamental to ensure convergence properties of the generated sequence $\{x_k\}$.

In a nonmonotone framework, we may have that

$$f(x_{k+1}) > f(x_k),$$

and hence, without some *control mechanism*, it may be not possible to guarantee the convergence of $\{f(x_k)\}$.

We provide here a rough idea of the conceptual and technical steps of the nonmonotone approaches later described in detail. In order to permit an increase of the objective function, condition (24.1) can be replaced, for instance, by a condition of the form

$$f(x_k + \alpha_k d_k) \leq W_k + \gamma \alpha_k \nabla f(x_k)^T d_k, \quad (24.2)$$

where $W_k \geq f(x_k)$ is the so-called *reference value*. The above condition refers to a *nonmonotone Armijo-type* line search. By a suitable definition of W_k related to the *past values* $f(x_{k-1}), f(x_{k-2}), \dots$, it is possible to ensure that the sequence $\{W_k\}$ is monotonically decreasing. Then, by assuming that a condition of the following form holds

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|), \quad (24.3)$$

where $\sigma : R^+ \rightarrow R^+$ is a forcing function, by an inductive reasoning it is possible to prove that the sequences $\{W_k\}$ and $\{f(x_k)\}$ converges to the same value. Note that (24.3) is satisfied, for instance, by (24.2) and suitable conditions on d_k . However, as we will see later, there are different ways to guarantee that (24.3) holds and to exploit it in a nonmonotone framework.

24.3 Reference Values and Preliminary Results

We give here a formal definition of a *reference value*, which is the value chosen for defining the *sufficient reduction* of some function f that can represent our objective function in an unconstrained problem or a *merit function* in some constrained problem.

Let $D \subseteq R^n$ be a given non empty set, let $x_0 \in D$ and let $f : D \rightarrow R$. Then we can define the level set relative to D and x_0 , that is:

$$\mathcal{L}_0 = \{x \in D : f(x) \leq f(x_0)\}.$$

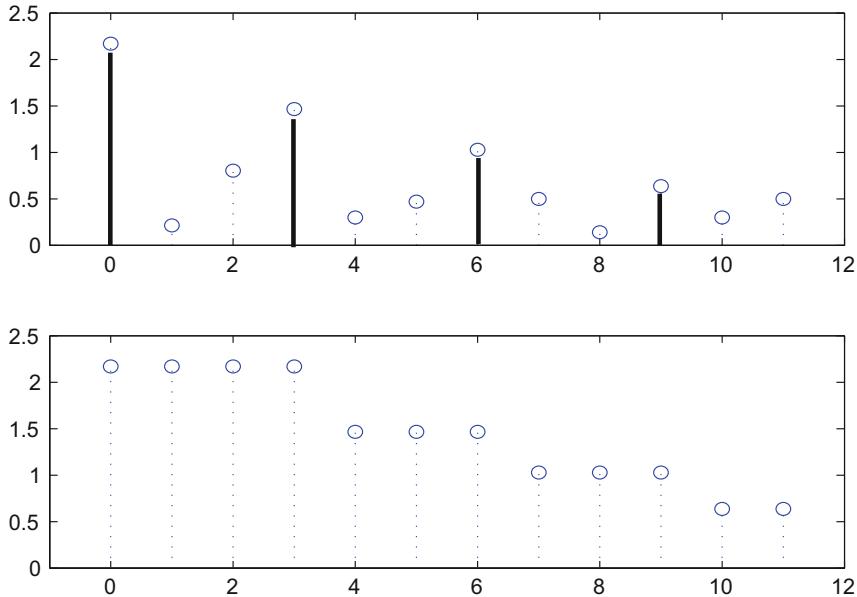


Fig. 24.1 Sequences $\{f(x_k)\}$ and $\{W_k\}$

If $\{x_k\}$ is a sequence of points $x_k \in D$ generated by some algorithm, one of the best known reference values, for $k = 0, 1, \dots$, is the number

$$W_k = \max_{0 \leq j \leq \min(k, M)} \{f(x_{k-j})\}, \quad (24.4)$$

where $M \geq 0$ is the *memory length*, which determines the number of past values of f taken into account. We obviously have that $f(x_k) \leq W_k$ and that $W_0 = f(x_0)$.

In Fig. 24.1 we show a short sequence of values $f(x_k)$, and the corresponding values of W_k for $M = 3$, which were generated by assuming that $f(x_{k+1}) < W_k$. We note that the sequence $\{f(x_k)\}$ is non monotone, while the sequence $\{W_k\}$ is monotonically non increasing. At each k we can find (at least) one point $x_{\ell(k)}$, with

$$k - \min(k, M) \leq \ell(k) \leq k,$$

where the maximum is attained in the evaluation of W_k , that is: $W_k = f(x_{\ell(k)})$.

In the example of Fig. 24.1 it easily seen that we have

$$x_{\ell(0)} = x_{\ell(1)} = x_{\ell(2)} = x_{\ell(3)} = x_0 \quad \text{and} \quad x_{\ell(4)} = x_{\ell(5)} = x_{\ell(6)} = x_3.$$

Now we will establish a condition of sufficient decrease, which is at the basis of the convergence proofs of many non monotone methods and follows, in essence,

from a simplified version of the proof given in [122]. Preliminarily, we establish the following lemma.

Lemma 24.1 *Let $D \subseteq R^n$ and let $\{x_k\}$ be a sequence of points such that $x_0 \in D$ and, for all $k = 0, 1, \dots$ we have*

$$f(x_{k+1}) \leq W_k, \quad x_{k+1} \in D, \quad (24.5)$$

where W_k is the reference value defined in (24.4), for some given $M \geq 0$. Then:

- (i) *the sequence $\{W_k\}$ is monotonically non increasing;*
- (ii) *$x_k \in \mathcal{L}_0$ for all k .*

Suppose that (24.5) holds with strict inequality for all k , that is $f(x_{k+1}) < W_k$, then, for all $k > M$:

- (iii) *$f(x_{k+1+s}) < W_k$ for all $s \geq 0$*
- (iv) *$f(x_{k+1+M+s}) < W_k$ for all $s \geq 0$.*

Proof Recalling that $W_k = f(x_{\ell(k)})$ and using (24.5), we can write

$$f(x_{k+1}) \leq f(x_{\ell(k)}). \quad (24.6)$$

Noting that $\min(k+1, M) \leq \min(k, M) + 1$, we have

$$\begin{aligned} f(x_{\ell(k+1)}) &= \max_{0 \leq j \leq \min(k+1, M)} [f(x_{k+1-j})] \leq \max_{0 \leq j \leq \min(k, M)+1} [f(x_{k+1-j})] \\ &= \max\{f(x_{\ell(k)}), f(x_{k+1})\} = f(x_{\ell(k)}), \end{aligned}$$

where the last equality follows from (24.6). This proves (i).

As $\{f(x_{\ell(k)})\}$ is monotonically non increasing and $x_{\ell(0)} = x_0$, we have $f(x_k) \leq f(x_0)$ for all k , and hence, as we have assumed that $x_k \in D$ for all k , we have $x_k \in \mathcal{L}_0$ and this establishes (ii).

Assume now that $f(x_{k+1}) < W_k$ for all k . Then, for every $s \geq 0$, taking (i) into account, we can write, for all $s \geq 0$:

$$f(x_{k+1+s}) < W_{k+s} \leq W_k,$$

which proves (iii). Then, by (iii) we have, in particular, for $k > M$:

$$W_{k+1+M} = \max_{0 \leq j \leq M} f(x_{k+1+M-j}) = \max_{0 \leq s \leq M} f(x_{k+1+s}) < W_k,$$

and hence, by (i), we obtain (iv). \square

Now we prove the following result.

Proposition 24.1 *Let $f : D \rightarrow R$ be bounded from below on \mathcal{L}_0 . Let $\{x_k\}$ be a sequence of points such that $x_0 \in D$ and, for all $k = 0, 1, \dots$ we have*

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|), \quad \text{and} \quad x_{k+1} \in D, \quad (24.7)$$

where $\sigma : R^+ \rightarrow R^+$ is a forcing function and W_k is the reference value defined in (24.4), for some given $M \geq 0$. Suppose that f is Lipschitz-continuous on \mathcal{L}_0 , that is, that there exists $L > 0$ such that, for all $x, y \in \mathcal{L}_0$, we have:

$$|f(x) - f(y)| \leq L\|x - y\|. \quad (24.8)$$

Then:

- (i) $x_k \in \mathcal{L}_0$ for all k ;
- (ii) the sequences $\{W_k\}$ and $\{f(x_k)\}$ converge to the same limit W_* ;
- (iii) we have $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$.

Proof By Lemma 24.1 we have that the sequence $\{W_k\}$ is monotonically non increasing and that $x_k \in \mathcal{L}_0$ for all k . As f is bounded below on \mathcal{L}_0 , the monotone sequence $\{W_k\} = \{f(x_{\ell(k)})\}$ has a limit W_* for $k \rightarrow \infty$. Letting j be an integer such that $1 \leq j \leq M + 1$, reasoning by induction on j we will show that:

$$\lim_{k \rightarrow \infty} \|x_{\ell(k)-j+1} - x_{\ell(k)-j}\| = 0, \quad (24.9)$$

$$\lim_{k \rightarrow \infty} f(x_{\ell(k)-j}) = \lim_{k \rightarrow \infty} f(x_{\ell(k)}), \quad (24.10)$$

where the sequences are considered for k sufficiently large to have $\ell(k) \geq k - M > 1$. If $j = 1$, using (24.7), where k is replaced by $\ell(k) - 1$, we have:

$$f(x_{\ell(k)}) \leq f(x_{\ell(\ell(k)-1)}) - \sigma(\|x_{\ell(k)} - x_{\ell(k)-1}\|). \quad (24.11)$$

Therefore, taking limits and recalling the definition of forcing function, by (24.11) and the convergence of $\{f(x_{\ell(k)})\}$ we obtain

$$\lim_{k \rightarrow \infty} \|x_{\ell(k)} - x_{\ell(k)-1}\| = 0. \quad (24.12)$$

Then, from (24.8), (24.12) and the fact that $\lim_{k \rightarrow \infty} f(x_{\ell(k)}) = W_*$, we obtain also that

$$\lim_{k \rightarrow \infty} f(x_{\ell(k)-1}) = \lim_{k \rightarrow \infty} f(x_{\ell(k)}) = W_*,$$

so that (24.9) and (24.10) hold at each k for $j = 1$.

Now suppose that (24.10) holds for a given j . By (24.7) we can write

$$f(x_{\ell(k)-j}) \leq f(x_{\ell(\ell(k)-j-1)}) - \sigma (\|x_{\ell(k)-j} - x_{\ell(k)-j-1}\|).$$

Taking limits for $k \rightarrow \infty$ and recalling (24.10) we obtain

$$\lim_{k \rightarrow \infty} \|x_{\ell(k)-j} - x_{\ell(k)-j-1}\| = 0,$$

which implies, together with (24.8) and (24.10),

$$\lim_{k \rightarrow \infty} f(x_{\ell(k)-j-1}) = \lim_{k \rightarrow \infty} f(x_{\ell(k)}).$$

From the preceding limits it follows that (24.9) and (24.10) hold when j is replaced by $j + 1$ and the induction is complete. It can be concluded that, for any given $j \in \{1, \dots, M + 1\}$ the limits (24.9) and (24.10) must hold.

Letting $L(k) = \ell(k + M + 1)$, in particular we have that (24.9) and (24.10) must be true when we replace $\ell(k)$ with $L(k)$. Moreover, for sufficiently large values of k , we can write:

$$\begin{aligned} x_{L(k)} &= x_k + (x_{k+1} - x_k) + \dots + (x_{L(k)} - x_{L(k)-1}) \\ &= x_k + \sum_{j=1}^{L(k)-k} (x_{L(k)-j+1} - x_{L(k)-j}). \end{aligned} \quad (24.13)$$

As $\ell(k + M + 1) \leq k + M + 1$, we have $L(k) - k \leq M + 1$, and hence (24.9) and (24.13) imply

$$\lim_{k \rightarrow \infty} \|x_k - x_{L(k)}\| = 0. \quad (24.14)$$

As $\{f(x_{\ell(k)})\}$ has a limit, by (24.8) and the convergence of $\{f(x_{\ell(k)})\}$ it follows that

$$\lim_{k \rightarrow \infty} f(x_k) = \lim_{k \rightarrow \infty} f(x_{L(k)}) = \lim_{k \rightarrow \infty} f(x_{\ell(k+M+1)}) = W_*,$$

which proves assertion (ii). Assertion (iii) follows from (24.7) and (ii). \square

Some immediate consequences of the preceding proposition will be given in the sequel. Preliminarily we recall the following result.

Proposition 24.2 *Let $f : R^n \rightarrow R$ be a continuous function and let $\{x_k\}$ be a sequence converging to $\bar{x} \in R^n$. Suppose there exists an infinite subsequence $\{x_k\}_K$, which is monotonically decreasing, that is, if $k_1, k_2 \in K$ with $k_2 > k_1$, we have $f(x_2) < f(x_1)$. Then, \bar{x} is not a local unconstrained maximizer of f .*

Proof Let $\{x_k\}$ be a sequence converging to \bar{x} . By continuity of f , we have

$$\lim_{k \rightarrow \infty} f(x_k) = f(\bar{x}).$$

Let now $\{f(x_k)\}_K$, the function values at the points of the subsequence $\{x_k\}_K$. As these values are strictly decreasing we must have $f(x_k) > f(\bar{x})$ for all $k \in K$. In fact, reasoning by contradiction, suppose there exists $\hat{k} \in K$ such that $f(x_{\hat{k}}) \leq f(\bar{x})$. Then, we can find $k_1 > \hat{k}$ in K such that all for $k \in K$ with $k > k_1$ we must have

$$f(x_k) < f(x_{k_1}) < f(x_{\hat{k}}) \leq f(\bar{x}),$$

so that, taking limits for $k \in K, k \rightarrow \infty$, we should obtain

$$f(\bar{x}) \leq f(x_{k_1}) < f(x_{\hat{k}}) \leq f(\bar{x}),$$

which yields a contradiction. Therefore, as $f(x_k) > f(\bar{x})$ for all $k \in K$, in every neighborhood of \bar{x} we can find a point of the subsequence $\{x_k\}_K$ where f is greater than $f(\bar{x})$ so that \bar{x} cannot be a local maximizer. \square

Now we can state the following consequence of Proposition 24.1.

Proposition 24.3 *Let $f : R^n \rightarrow R$ be a continuously differentiable function on R^n , let $x_0 \in R^n$ and assume that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. Let $\{x_k\}$ be a sequence of points in R^n such that condition (24.7) is satisfied (with $D = R^n$) for all $k = 0, 1, \dots$, for some given $M \geq 0$.*

Then the assertions of Proposition 24.1 hold with $D = R^n$. Moreover, the sequence $\{x_k\}$ admits limit points and no limit point is a local unconstrained maximizer of f .

Proof It is easily seen that the assumptions of Proposition 24.1 are satisfied by letting $D = \mathbb{R}^n$. In particular, for all $x, y \in \mathcal{L}_0$ we can write, using the Theorem of the Mean:

$$|f(x) - f(y)| \leq \max_{z \in C} \|\nabla f(z)\| \|x - y\|,$$

where C is a compact convex set containing \mathcal{L}_0 . This shows that f is Lipschitz continuous on C , with Lipschitz constant $L = \max_{z \in C} \|\nabla f(z)\|$. Thus, all the assertions of Proposition 24.1 are valid. Therefore, as $x_k \in \mathcal{L}_0$, the compactness of \mathcal{L}_0 implies the existence of limit points in the level set.

Suppose now that \bar{x} is one of these limit points, let $\{x_k\}_K$ be a subsequence converging to \bar{x} and suppose $k > M$. As $\|x_k - x_{k-1}\| \rightarrow 0$ and $k - \ell(k) \leq M$, we have that $\|x_k - x_{\ell(k)}\| \rightarrow 0$ and hence we have also that

$$\lim_{k \in K, k \rightarrow \infty} x_{\ell(k)} = \bar{x},$$

so that

$$\lim_{k \in K, k \rightarrow \infty} W_k \equiv \lim_{k \in K, k \rightarrow \infty} f(x_{\ell(k)}) = f(\bar{x}).$$

Now, recalling (iv) of Lemma 24.1, we can construct a subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$, converging to \bar{x} , such that $\{W_k\}_{K_1}$ is strictly decreasing. Then the assertion follows from Proposition 24.2. \square

24.4 Armijo-Type Nonmonotone Line Searches

We consider here the nonmonotone version of a backtracking Armijo-type line search algorithm that was introduced in connection with unconstrained minimization methods. We suppose that a sequence $\{x_k\}$ is generated, according to the iteration $x_{k+1} = x_k + \alpha_k d_k$, where α_k is the step-size and d_k is a search direction, computed at each step k , on the basis of some local model.

We already know that the line search algorithms are based on conditions that guarantee

- (i) a sufficient decrease of the objective function;
- (ii) a sufficiently large step-size.

In some cases and, in particular, in the case of nonmonotone methods, it is also required to enforce the limit

$$\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0. \quad (24.15)$$

Nonmonotone line search methods essentially have the same structure of monotone methods, but impose a condition of sufficient decrease with respect to a reference value (and not to the current value $f(x_k)$) and also guarantee that the preceding limit holds.

Under the assumption that f is continuously differentiable, a first example is a (purely conceptual!) version of nonmonotone Armijo's method, reported below, where

- the initial tentative step-size is a constant step-size, that is $\Delta_k = a > 0$ for all k ;
- d_k is a descent direction such that $\nabla f(x_k)^T d_k < 0$;
- W_k is defined as in (24.4).

Algorithm 24.1 (Nonmonotone Armijo's Method)

Data: $\gamma \in (0, 1/2)$, $\delta \in (0, 1)$, $\Delta_k = a > 0$.

Set $\alpha = a$.

While $f(x_k + \alpha d_k) > W_k + \gamma \alpha \nabla f(x_k)^T d_k$
 set $\alpha = \delta \alpha$.

End While

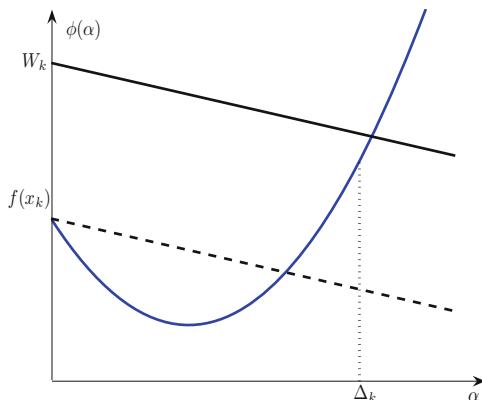
Set $\alpha_k = \alpha$ and **exit**.

The finite termination of this algorithm follows easily from the corresponding result established in the monotone case (see Proposition 10.2), noting that now we have $f(x_k) \leq W_k$.

In Fig. 24.2 we plot the behaviour of $\phi(\alpha) = f(x_k + \alpha d_k)$ and we show the relaxation of the descent requirement that we have when $W_k > f(x_k)$.

The initial tentative step-size $\Delta_k = a$ is accepted, even if it corresponds to an increase of f with respect to $f(x_k)$, provided that $f(x_k + ad_k)$ does not exceed the value $W_k + \gamma a \nabla f(x_k)^T d_k$.

Fig. 24.2 Nonmonotone acceptance criterion



On the basis of the results established in the preceding section, we can derive a convergence proof for a nonmonotone version of the standard backtracking Armijo's method with initial step-size $\Delta_k = a$, by imposing suitable conditions on the search direction that guarantee satisfaction of condition (24.7) and also ensure that the step-size is sufficiently large.

Proposition 24.4 (Convergence of Nonmonotone Armijo's Method) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set*

$$\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$$

is compact. Assume that, for every k , we have $\nabla f(x_k) \neq 0$ and that d_k satisfies the following conditions:

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^p, \quad p > 0, \quad (24.16)$$

$$\|d_k\|^q \leq c_2 \|\nabla f(x_k)\|, \quad q > 0 \quad (24.17)$$

where $c_1, c_2 > 0$ and $pq > 1$. Then the nonmonotone Armijo's method of Algorithm 24.1 with initial step-size $\Delta_k = a$ determines in a finite number of steps an $\alpha_k > 0$ such that the sequence defined by $x_{k+1} = x_k + \alpha_k d_k$ satisfies the following conditions

- (a₁) $x_k \in \mathcal{L}_0$ for all k ;
- (a₂) the sequences $\{f(x_k)\}$ and $\{W_k\}$ converge to the same limit;
- (a₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (a₄) $\lim_{k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$.

Proof From the acceptance rule

$$f(x_k + \alpha_k d_k) \leq W_k + \gamma \alpha_k \nabla f(x_k)^T d_k,$$

by (24.16) and (24.17), we obtain

$$f(x_k + \alpha_k d_k) \leq W_k - \gamma \frac{c_1}{c_2^p} \alpha_k \|d_k\|^{pq}.$$

Moreover, as $\alpha_k \leq a$, we have $\alpha_k/a \leq 1$, and hence we can write

$$f(x_k + \alpha_k d_k) < W_k - \gamma \frac{c_1}{c_2^p} \alpha_k \left(\frac{\alpha_k}{a}\right)^{pq-1} \|d_k\|^{pq} = W_k - \gamma \frac{c_1}{c_2^p a^{pq-1}} \|\alpha_k d_k\|^{pq}.$$

Therefore, we have

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|), \quad (24.18)$$

where σ is the forcing function

$$\sigma(t) = \gamma \frac{c_1}{c_2^p a^{pq-1}} t^{pq}.$$

Thus, condition (24.7) is satisfied and hence, by the compactness of the level set \mathcal{L}_0 , it can be easily verified that the assumptions of Proposition 24.1 are satisfied, so that conditions (a₁), (a₂) and (a₃) follow from this proposition.

Moreover, as \mathcal{L}_0 is compact, using (24.16) and employing the same arguments used in Chap. 10 for establishing (10.24) we have that the forcing function $\sigma_0(t) \equiv c_1 at^p/M$, where M is an upper bound of $\|\nabla f(x)\|$ on \mathcal{L}_0 , satisfies the condition

$$a \geq \frac{1}{\|d_k\|} \sigma_0 \left(\left| \nabla f(x_k)^T d_k \right| / \|d_k\| \right). \quad (24.19)$$

Thus, we can repeat essentially the same reasonings employed in the proof of Proposition 10.3, taking into account (a₃) and noting that, if $\alpha_k < a$, we have:

$$f(x_k + \frac{\alpha_k}{\delta} d_k) > W_k + \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k \geq f(x_k) + \gamma \frac{\alpha_k}{\delta} \nabla f(x_k)^T d_k, \quad (24.20)$$

so that condition (10.19) must hold. \square

We can note that the conditions on d_k are satisfied if $d_k = -\nabla f(x_k)$ and we set $c_1 = 1$, $p = 2$, $q = 1$, $c_2 = 1$. This implies that we get immediately a convergence result for a nonmonotone version of the gradient method.

24.5 Nonmonotone Armijo-Goldstein and Parabolic Searches

In alternative to the Armijo-type backtracking algorithm defined in the preceding section, we can also consider nonmonotone versions of linesearch algorithms that permit an increase of the initial tentative step-size and yet guarantee satisfaction of the condition (24.7). This can be useful when the choice of the initial tentative step-size is not dictated by the properties of the local model that yields d_k .

A first possibility is the definition of a nonmonotone extension of the acceptability conditions of a (modified) Armijo-Goldstein algorithm. In the nonmonotone case we must again impose suitable conditions on the search direction, as in Proposition 24.4 and also, as we do not perform backtracking and admit an increase of the initial step-size, we must place some bound on the maximal step-size.

Another possibility is that of defining a nonmonotone *norm reducing* procedure, based on the *parabolic search* considered in Chap. 10, which directly yields an acceptability condition satisfying (24.7).

In the monotone case, the acceptability condition along a descent direction was a condition of the form

$$f(x_k + \alpha d_k) \leq f(x_k) - \gamma \alpha^2 \|d_k\|^2$$

and in the nonmonotone case $f(x_k)$ is replaced by the reference value W_k . This rule does not require a gradient evaluation in the acceptance condition and hence it can be adapted also, as we will see, to the construction of derivative-free linesearches. Following [134], the two approaches considered above will be combined into a single scheme in order to avoid the repetition of similar arguments.

Let $x_k \in R^n$ be a point generated by some algorithm and let $d_k \neq 0$ a descent direction for f at x_k that satisfies $\nabla f(x_k)^T d_k < 0$. We define a condition of sufficient decrease in the following form:

$$f(x_k + \alpha d_k) \leq W_k + \gamma_1 \alpha \nabla f(x_k)^T d_k - \gamma_2 \alpha^2 \|d_k\|^2, \quad (24.21)$$

where W_k is defined by (24.4) and γ_1, γ_2 satisfy the conditions:

$$1 > \gamma_1 \geq 0, \quad \gamma_2 \geq 0 \quad \gamma_1 + \gamma_2 > 0. \quad (24.22)$$

When $\gamma_2 = 0$ we must require that an upper bound \bar{a} on the step-size is imposed and hence we assume that $\Delta_k < \bar{a}$. To simplify notation, we set $f_k = f(x_k)$, $\nabla f_k = \nabla f(x_k)$ and we define $\beta_k = \sigma_0(|\nabla f_k^T d_k|/\|d_k\|)$, where σ_0 is a forcing function.

Now we can define the following (conceptual) algorithm model.

Algorithm 24.2 (Nonmonotone Line Search Model)

Data: γ_1, γ_2 satisfying (24.22), $\gamma_1 < \tilde{\gamma}_1 < 1, \gamma_2 < \tilde{\gamma}_2, 0 < \delta_1 \leq \delta_2 < 1,$
 $\tau_2 > \tau_1 > 1$; if $\gamma_2 = 0$, $\bar{a} \in R$ with $\bar{a} > 0$, else $\bar{a} = \infty$.

Step 1. Choose Δ_k such that $0 < \Delta_k \leq \bar{a}$, and set $\alpha = \Delta_k, j = 0$ and $h = 0$.

Step 2.

While

$f(x_k + \alpha d_k) > W_k + \gamma_1 \alpha \nabla f_k^T d_k - \gamma_2 \alpha^2 \|d_k\|^2$
choose $\delta \in [\delta_1, \delta_2]$ and set $\alpha = \delta \alpha, j = j + 1$

End While

Step 3. If $j \geq 1$ set $\alpha_k = \alpha, \delta_k = \delta$ and **terminate**.

Step 4. If $\alpha \|d_k\| \geq \beta_k$ or $f(x_k + \alpha d_k) > f_k$ then set $\alpha_k = \alpha$ and **terminate**;
else choose $\tau \in [\tau_1, \tau_2]$.

(continued)

Algorithm 24.2 (continued)**Step 5.****While** $\tau\alpha < \bar{a}$ and:

$$\begin{aligned} f(x_k + \alpha d_k) &< f_k + \tilde{\gamma}_1 \alpha \nabla f_k^T d_k, \quad \gamma_1 > 0 \\ f(x_k + \alpha d_k) &< f_k - \tilde{\gamma}_2 \alpha^2 \|d_k\|^2, \quad \gamma_2 > 0 \\ f(x_k + \tau\alpha d_k) &< \min\{f(x_k + \alpha d_k), f_k + \gamma_1 \tau \alpha \nabla f(x_k)^T d_k - \gamma_2 (\tau \alpha)^2 \|d_k\|^2\} \end{aligned}$$

set $\alpha = \tau\alpha$, $h = h + 1$ and choose $\tau \in [\tau_1, \tau_2]$.

End While**Step 6.** Set $\alpha_k = \alpha$, $\tau_k = \tau$ and **terminate**.

The next proposition shows that the algorithm is well defined.

Proposition 24.5 Suppose that f is bounded below on R^n and that $\nabla f(x_k)^T d_k < 0$. Then the nonmonotone linesearch algorithm 24.2 determines in a finite number of steps a step-size $\alpha_k > 0$ such that

$$f(x_k + \alpha_k d_k) \leq W_k + \gamma_1 \alpha_k \nabla f_k^T d_k - \gamma_2 \alpha_k^2 \|d_k\|^2,$$

and at least one of the following conditions holds:

$$\alpha_k < \Delta_k \quad \text{and} \quad f(x_k + \frac{\alpha_k}{\delta_k} d_k) > f_k + \gamma_1 \frac{\alpha_k}{\delta_k} \nabla f_k^T d_k - \gamma_2 (\frac{\alpha_k}{\delta_k})^2 \|d_k\|^2, \quad (24.23)$$

$$\alpha_k = \Delta_k \quad \text{and} \quad \alpha_k \|d_k\| \geq \beta_k \quad (24.24)$$

$$\alpha_k = \Delta_k \quad \text{and} \quad f(x_k + \alpha_k d_k) \geq f_k \quad (24.25)$$

$$\gamma_2 = 0 \quad \text{and} \quad \tau_k \alpha_k > \bar{a} \quad (24.26)$$

$$\gamma_1 > 0 \quad \text{and} \quad f(x_k + \alpha_k d_k) \geq f_k + \tilde{\gamma}_1 \alpha_k \nabla f_k^T d_k, \quad (24.27)$$

$$\gamma_2 > 0 \quad \text{and} \quad f(x_k + \alpha_k d_k) \geq f_k - \tilde{\gamma}_2 \alpha_k^2 \|d_k\|^2, \quad (24.28)$$

$$f(x_k + \tau_k \alpha_k d_k) \geq \min\{f(x_k + \alpha_k d_k), f_k + \gamma_1 \tau_k \alpha_k \nabla f_k^T d_k - \gamma_2 (\tau_k \alpha_k)^2 \|d_k\|^2\}. \quad (24.29)$$

Proof We must show that the algorithm terminates, by proving that the while cycles at Step 3 and Step 6 terminate in a finite number of steps. If the cycle at Step 3 does

not terminate, we have that $0 < \alpha(j) \leq \Delta_k \delta_u^j$, where j is the counter of the inner steps of the cycle, and that

$$\frac{f(x_k + \alpha(j)d_k) - f_k}{\alpha(j)} > \gamma_1 \nabla f_k^T d_k - \gamma_2 \alpha(j) \|d_k\|^2.$$

Thus, taking limits for $j \rightarrow \infty$ and noting that $\alpha(j) \rightarrow 0$ and $\gamma_1 < 1$ we obtain $\nabla f_k^T d_k \geq 0$, which contradicts our assumption. Suppose now that the while cycle at Step 6 does not terminate. Letting h be a counter of the inner iterations, we have $\alpha(h) \geq \tau_l^h \Delta_k$ and hence $\alpha(h) \rightarrow \infty$: this implies that $\gamma_2 > 0$ and also that $f(x_k + \alpha(h)d_k) < W_k - \tilde{\gamma}_2 \alpha(h)^2 \|d_k\|^2$, so that $f(x_k + \alpha(h)d_k) \rightarrow -\infty$, which contradicts our assumptions. Then the algorithm is well defined and the assertions (24.23)–(24.29) are immediate consequences of the conditions for termination given in the algorithm, taking into account the fact that $W_k \geq f_k$. \square

Now we can establish the convergence properties of Algorithm 24.2. In particular, we consider the case when the algorithm is employed only at a subsequence $\{x_k\}_K$ of points while at the remaining points x_k with $k \notin K$ the point x_{k+1} is generated with some unspecified rule in a way that condition (24.7) is satisfied.

Proposition 24.6 (Convergence of Algorithm 24.2) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that $\nabla f(x_k) \neq 0$ for all k and assume that:*

- (i) *there exists an infinite subsequence $\{x_k\}_K$ such that, for every $k \in K$, we have that $x_{k+1} = x_k + \alpha_k d_k$, where d_k is a descent direction such that $\nabla f(x_k)^T d_k < 0$, and α_k is computed with Algorithm 24.2 using the reference value W_k defined in (24.4);*
- (ii) *at each x_k , $k \in K$, if we have $\gamma_2 = 0$ in Algorithm 24.2, then d_k satisfies conditions (24.16) and (24.17);*
- (iii) *there exists a forcing function $\sigma_a : R^+ \rightarrow R^+$ such that*

$$f(x_{k+1}) \leq W_k - \sigma_a(\|x_{k+1} - x_k\|),$$

at every x_k such that $k \notin K$.

Then, we have

- (c₁) $x_k \in \mathcal{L}_0$ for all k ;
- (c₂) *the sequences $\{f(x_k)\}$ and $\{W_k\}$ converge to the same limit;*
- (c₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c₄) $\lim_{k \in K, k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$.

Proof First we show that the assumptions of Proposition 24.1 are satisfied. If $k \in K$ we can distinguish the case $\gamma_2 = 0$ and $\gamma_2 > 0$.

If $\gamma_2 = 0$ then $\gamma_1 > 0$, and hence, reasoning as in the proof of Proposition 24.4 and taking into account the fact that $\alpha_k \leq \bar{a} < \infty$, we have:

$$f(x_{k+1}) \leq W_k - \sigma_b(\|x_{k+1} - x_k\|), \quad (24.30)$$

where σ_b is the forcing function

$$\sigma_b(t) = \gamma_1 \frac{c_1}{c_2^p \bar{a}^{pq-1}} t^{pq},$$

If $k \in K$ and $\gamma_2 > 0$ we have

$$f(x_{k+1}) \leq W_k - \sigma_c(\|x_{k+1} - x_k\|), \quad (24.31)$$

where $\sigma_c(t) = \gamma_2 t^2$. Then, for every k we have that the assumptions of Proposition 24.1 are satisfied, provided that we choose $\sigma(t) = \min\{\sigma_a(t), \sigma_b(t)\}$ if $\gamma_2 = 0$ and $\sigma(t) = \min\{\sigma_a(t), \sigma_c(t)\}$ if $\gamma_2 > 0$.

Therefore, assertions (c₁), (c₂) and (c₃) follow from this proposition. In order to prove (c₄), let us assume, by contradiction, that (c₄) is false. By (c₁) and the compactness of \mathcal{L}_0 , we can find an infinite subsequence with index set $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \rightarrow \infty} x_k = \bar{x},$$

and

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} < 0. \quad (24.32)$$

Now, suppose first that Algorithm 24.2, for an infinite subsequence, which we relabel $\{x_k\}_{K_1}$, terminates with $\alpha_k < \Delta_k$, so that (24.23) holds with $\delta_k \in [\delta_i, \delta_u]$, that is:

$$f(x_k + \frac{\alpha_k}{\delta_k} d_k) > f_k + \gamma_1 \frac{\alpha_k}{\delta_k} \nabla f_k^T d_k - \gamma_2 \left(\frac{\alpha_k}{\delta_k} \right)^2 \|d_k\|^2.$$

Then, by the Mean Value Theorem there must exist a point $u_k = x_k + \xi_k (\alpha_k / \delta_k) d_k$ with $\xi_k \in (0, 1)$, such that

$$\nabla f(u_k)^T d_k \geq \gamma_1 \nabla f(x_k)^T d_k - \gamma_2 \frac{\alpha_k}{\delta_k} \|d_k\|^2. \quad (24.33)$$

Now, by (c₃) we have that $u_k \rightarrow \bar{x}$ for $k \in K_1, k \rightarrow \infty$ and hence, dividing both members of (24.33) by $\|d_k\|$, taking limits and taking into account the fact that $\gamma_1 < 1$, we get $\nabla f(\bar{x})^T \bar{d} \geq 0$, which contradicts (24.32).

We can repeat essentially the same reasoning, (with reference to some infinite subsequence of $\{x_k\}_{K_1}$) by employing again the Mean Value Theorem, in correspondence to cases (24.25), (24.27), (24.28) and (24.29).

If (24.24) holds for an infinite subsequence of $\{x_k\}_{K_1}$, recalling (c₃) and taking limits, we get immediately a contradiction to (24.32).

Finally, suppose that (24.26) holds for a subsequence, say again $\{x_k\}_{K_1}$. Then, as $\gamma_2 = 0$ we have that $f(x_{k+1}) \leq W_k + \gamma_1 \alpha_k \nabla f_k^T d_k$, and hence by (c₃) we have in the limit $\lim_{k \in K_1, k \rightarrow \infty} \alpha_k \nabla f(x_k)^T d_k = 0$. Therefore, as $\alpha_k > \bar{a}/\tau_u$ we have

$$\lim_{k \in K_1, k \rightarrow \infty} \nabla f(x_k)^T d_k = 0,$$

so that, by condition (24.16), we have $\lim_{k \in K_1, k \rightarrow \infty} \nabla f(x_k) = 0$. This implies, for some subsequence, say again $\{x_k\}_{K_1}$:

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} = 0,$$

and hence we get a contradiction to (24.32). \square

Remark 24.1 We can note that if we set $\gamma_2 = 0$ and $\Delta_k = \bar{a}$, then we obtain for $k \in K$ an extension of the nonmonotone backtracking Armijo's method to the case in which the linesearch is (possibly) performed only at a subsequence. \square

24.6 Nonmonotone Derivative-Free Linesearches

In this section, in connection with unconstrained minimization problems, we describe nonmonotone linesearches that extend the results established in Chap. 19 for monotone derivative-free linesearches, by introducing the reference value W_k defined earlier.

As we know, if derivative information is not available we cannot establish analytically whether the search direction d_k is a descent direction and hence the line search can terminate with a failure with $\alpha_k = 0$.

A non monotone version of an Armijo-type *parabolic* line search is stated in the following scheme.

Algorithm 24.3 (Nonmonotone Derivative-Free Armijo-Type Line-search)

Data. $d_k \neq 0$, W_k defined as in (24.4); parameters:

$$a > 0, \gamma > 0, \rho_k \in (0, 1), 0 < \delta_l \leq \delta_u < 1.$$

1. Choose $a_k \geq a/\|d_k\|$ and set $\alpha = a_k$, $\xi_k = \rho_k \min[1, a_k, 1/\|d_k\|]$.

2. While $f(x_k \pm \alpha d_k) > W_k - \gamma \alpha^2 \|d_k\|^2$ do

If $\alpha \|d_k\| < \xi_k$ then

set $\alpha_k = 0$, $\eta_k = \alpha$ and **exit**.

Else

choose $\delta \in [\delta_l, \delta_u]$ and set $\alpha = \delta \alpha$.

End If

End while

3. Set $\alpha_k = u\alpha$, where $u \in \{-1, 1\}$ is the value for which the condition of sufficient reduction of Step 2 is satisfied and **exit**. \square

It can be easily proved that the algorithm terminates.

Proposition 24.7 Suppose that $d_k \neq 0$. Then Algorithm 24.3 determines in a finite number of steps a step-size α_k such that

$$f(x_k + \alpha_k d_k) \leq W_k - \gamma \alpha_k^2 \|d_k\|^2, \quad (24.34)$$

and at least one of the following conditions holds:

$$\alpha_k = 0 \quad \text{and} \quad f(x_k \pm \eta_k d_k) > W_k - \gamma \eta_k^2 \|d_k\|^2, \quad \eta_k < \xi_k \quad (24.35)$$

$$0 < |\alpha_k| = a_k \quad \text{and} \quad |\alpha_k| \geq a/\|d_k\|, \quad (24.36)$$

$$0 < |\alpha_k| < a_k \quad \text{and} \quad f(x_k \pm \frac{\alpha_k}{\delta_k} d_k) > f_k - \gamma (\frac{\alpha_k}{\delta_k})^2 \|d_k\|^2, \quad \delta_k \in [\delta_l, \delta_u] \quad (24.37)$$

Proof The while cycle at Step 2 terminates in a finite number of inner iterations because $\alpha \leq \delta_u \alpha$ with $\delta_u < 1$ at each inner step and hence the condition $\alpha \|d_k\| < \xi$ will be satisfied in a finite number of steps. If the algorithm terminates we may have either that $\alpha_k = 0$ or that $\alpha_k \neq 0$. In both cases condition (24.34) is satisfied. In the first case the scalar $\eta_k > 0$ is such that (24.35) is satisfied. When $\alpha_k \neq 0$ termination may occur either with $|\alpha_k| = a_k$ if the initial step-size is accepted (with an appropriate sign), and hence (24.36) holds, or with $0 < |\alpha_k| < a_k$, when the

initial tentative step-size has been reduced at least one time and therefore (24.37) must be satisfied. \square

Now, we can establish the convergence properties of Algorithm 24.3, under the assumption that the algorithm is employed for an infinite subsequence.

Proposition 24.8 (Convergence of Algorithm 24.3) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that:*

- (i) *there exists an infinite subsequence $\{x_k\}_K$ such that, for every $k \in K$, we have that $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ and the step-size α_k is computed by employing Algorithm 24.3 with the reference value W_k defined as in (24.4);*
- (ii) *we have $\rho_k \rightarrow 0$ for $k \in K, k \rightarrow \infty$;*
- (iii) *there exists a forcing function $\sigma_a : R^+ \rightarrow R^+$ such that*

$$f(x_{k+1}) \leq W_k - \sigma_a(\|x_{k+1} - x_k\|) \quad (24.38)$$

at every x_k such that $k \notin K$.

Then, we have

- (c₁) $x_k \in \mathcal{L}_0$ for all k ;
- (c₂) *the sequences $\{f(x_k)\}$ and $\{W_k\}$ converge to the same limit;*
- (c₃) $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$;
- (c₄) $\lim_{k \in K, k \rightarrow \infty} \nabla f(x_k)^T d_k / \|d_k\| = 0$.

Proof Taking into account the compactness of \mathcal{L}_0 , the assumption that f is continuously differentiable, the inequality (24.34) and the condition (24.38), it is easily seen that the assumptions of Proposition 24.1 are satisfied. Therefore, assertions (c₁), (c₂) and (c₃) follow from this proposition.

In order to prove (c₄), let us assume, by contradiction, that (c₄) is false. By (c₁) and the compactness of \mathcal{L}_0 , we can find an infinite subsequence with index set $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \rightarrow \infty} x_k = \bar{x},$$

and

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} \neq 0, \quad \text{where } \|\bar{d}\| = 1. \quad (24.39)$$

Now, suppose first that there exists $\bar{k} \in K_1$ such that, for all $k \in K_1, k \geq \bar{k}$, Algorithm 24.3 terminates with $\alpha_k = 0$. In this case we have that (24.35) must be satisfied.

From (24.35), using the Mean Value Theorem, we have that there exist points $u_k = x_k + \zeta_k \eta_k d_k$ and $v_k = x_k - \beta_k \eta_k d_k$, with $\zeta_k, \beta_k \in (0, 1)$, such that

$$\frac{\nabla f(u_k)^T d_k}{\|d_k\|} > -\gamma_1 \eta_k \|d_k\| \quad \frac{\nabla f(v_k)^T d_k}{\|d_k\|} < \gamma_1 \eta_k \|d_k\| \quad (24.40)$$

As $\eta_k \|d_k\| \leq \rho_k$ and $\rho_k \rightarrow 0$ for $k \in K_1, k \rightarrow \infty$, we have that $u_k \rightarrow \bar{x}$ and $v_k \rightarrow \bar{x}$, so that we get in the limit $\nabla f(\bar{x})^T \bar{d} = 0$ and this contradicts (24.48).

Therefore there must exist an infinite subset $K_2 \subseteq K_1$ such that $\alpha_k \neq 0$ for $k \in K_2$. Recalling the preceding proposition, we can distinguish two different cases.

Case (a) We have $0 < |\alpha_k| = a_k$ for an infinite subsequence, so that $|\alpha_k| \geq a/\|d_k\|$.

However, by (c₃) we must have $\alpha_k \|d_k\| \rightarrow 0$ and hence we get a contradiction.

Case (b) We have $0 < |\alpha_k| < a_k$ for an infinite subsequence. Then condition (24.36) must be satisfied. Then we can repeat a similar reasoning to that followed in the case $\alpha_k = 0$, using the Mean Value Theorem, replacing η_k with α_k/δ_k and taking into account assertion (c₃). Thus we get again a contradiction to (24.39).

□

We consider now the nonmonotone version of a derivative-free Armijo-Goldstein algorithm. The monotone version can be obtained as a special case by setting $M = 0$.

Algorithm 24.4 (Nonmonotone Derivative-Free Armijo-Goldstein-Type Linesearch)

Data. $d_k \neq 0$, W_k defined as in (24.4); parameters:

$$\alpha_k > 0, \gamma_2 > \gamma_1 > 0, \rho_k \in (0, 1), 0 < \delta_l \leq \delta_u < 1, 1 < \tau_1 < \tau_2.$$

1. Set $\alpha = a_k$ and $\xi = \rho_k \min[1, a_k, 1/\|d_k\|]$.
2. **While** $f(x_k \pm \alpha d_k) > W_k - \gamma_1 \alpha^2 \|d_k\|^2$ **do**
 - If** $\alpha \|d_k\| < \xi$ **then**
 - set $\alpha_k = 0, \eta_k = \alpha, \xi_k = \xi$ and **exit**.
 - Else**
 - choose $\delta \in [\delta_l, \delta_u]$ and set $\alpha = \delta \alpha$.
 - End If**
 - End while**
3. Let $t \in \{-1, 1\}$ be such that $f(x_k + t \alpha d_k) \leq W_k - \gamma_1 \alpha^2 \|d_k\|^2$ and set $\alpha = t \alpha$

(continued)

Algorithm 24.4 (continued)

4. If $|\alpha| < a_k$ set $\alpha_k = \alpha$ and **terminate**.
 5. If $f(x_k \pm \alpha d_k) \geq f(x_k)$, then set $\alpha_k = \alpha$ and **terminate**.
 6. Let $s \in \{-1, 1\}$ be such that $f(x_k + s\alpha d_k) < f(x_k)$, set $\alpha = s\alpha$ and choose $\tau \in [\tau_1, \tau_2]$
 7. **While**
- $$f(x_k + \alpha d_k) < f(x_k) - \gamma_2 \alpha^2 \|d_k\|^2,$$
- $$f(x_k + \tau \alpha d_k) < \min \left\{ f(x_k + \alpha d_k), f(x_k) - \gamma_1 (\tau \alpha)^2 \|d_k\|^2 \right\}$$
- set $\alpha = \tau \alpha$ and choose $\tau \in [\tau_1, \tau_2]$.
- End while**
8. Set $\alpha_k = \alpha$, $\tau_k = \tau$ and **terminate**.

□

The next proposition, which is the analogue of Proposition 24.7, shows that the algorithm terminates in a finite number of inner steps.

Proposition 24.9 Suppose that f is bounded below on R^n and that $d_k \neq 0$. Then Algorithm 24.4 determines in a finite number of steps and computes a step-size α_k such that

$$f(x_k + \alpha_k d_k) \leq W_k - \gamma_1 \alpha_k^2 \|d_k\|^2, \quad (24.41)$$

and at least one of the following conditions holds:

$$\alpha_k = 0 \text{ and } f(x_k \pm \eta_k d_k) > W_k - \gamma_1 \eta_k^2 \|d_k\|^2, \quad \eta_k < \xi_k \quad (24.42)$$

$$\alpha_k \neq 0 \text{ and } f(x_k \pm \frac{\alpha_k}{\delta_k} d_k) > f_k - \gamma_1 (\frac{\alpha_k}{\delta_k})^2 \|d_k\|^2, \quad \delta_k \in [\delta_l, \delta_u] \quad (24.43)$$

$$\alpha_k \neq 0 \text{ and } f(x_k \pm \alpha_k d_k) \geq f_k \quad (24.44)$$

$$\alpha_k \neq 0 \text{ and } f(x_k + \alpha_k d_k) \geq f_k - \gamma_2 \alpha_k^2 \|d_k\|^2, \quad f(x_k + \alpha_k d_k) < f_k \quad (24.45)$$

$$\alpha_k \neq 0 \text{ and } f(x_k + \tau_k \alpha_k d_k) \geq \min\{f(x_k + \alpha_k d_k), f_k - \gamma_1 \tau_k^2 \alpha_k^2 \|d_k\|^2\}$$

with $f(x_k + \alpha_k d_k) < f_k, \quad \tau_k \in [\tau_l, \tau_u].$ (24.46)

Proof We show that the while cycles at Step 2 and Step 7 terminate in a finite number of inner iterations. The cycle at Step 2 terminates because $\alpha \leq \delta_u \alpha$ with $\delta_u < 1$ at each inner step and hence the condition $\alpha \|d_k\| < \xi$ will be satisfied in a finite number of steps. The cycle at Step 7 terminates because, otherwise, we will have that $|\alpha| \rightarrow \infty$ and $f(x_k + \alpha d_k) \rightarrow -\infty$, which contradicts the assumption that f is bounded below.

If the algorithm terminates we may have either that $\alpha_k = 0$ or that $\alpha_k \neq 0$. In both cases condition (24.41) is satisfied. In the first case the scalar $\eta_k > 0$ is such that (24.42) is satisfied. When $\alpha_k \neq 0$ termination may occur at Steps 4,5 or 8. If termination occurs at Step 4 this implies that the initial tentative step-size has been reduced at least one time and hence (24.43) must be true. If the algorithm terminates at Step 5 we have $|\alpha_k| = a_k > 0$ and (24.44) is true. Finally, suppose that the algorithm terminates at Step 8. We can observe that the while cycle at step 7 is started only when $f(x_k + \alpha d_k) < f_k$ and, subsequently, the instructions at step 7 ensure that f is strictly decreasing for increasing values of $|\alpha|$, as $f(x_k + \tau \alpha d_k) < f(x_k + \alpha d_k)$. Thus, one of the conditions (24.45) or (24.46) must be true. \square

Now, we can establish the convergence properties of Algorithm 24.4, under the same assumptions considered for the Armijo-type Algorithm 24.3.

Proposition 24.10 (Convergence of Algorithm 24.4) *Let $f : R^n \rightarrow R$ be continuously differentiable on R^n and suppose that the level set $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. For $k = 0, 1, \dots$ let $\{x_k\}$ be a sequence such that:*

- (i) *there exists an infinite subsequence $\{x_k\}_K$ such that, for every $k \in K$, we have that*

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

where $d_k \neq 0$ and the step-size α_k is computed by employing Algorithm 24.4 with the reference value W_k defined in (24.4);

- (ii) *we have $\rho_k \rightarrow 0$ for $k \in K$, $k \rightarrow \infty$;*
- (iii) *there exists a forcing function $\sigma_a : R^+ \rightarrow R^+$ such that*

$$f(x_{k+1}) \leq W_k - \sigma_a(\|x_{k+1} - x_k\|) \quad (24.47)$$

at every x_k such that $k \notin K$.

Then, we have

- (c₁) *$x_k \in \mathcal{L}_0$ for all k ;*
- (c₂) *the sequences $\{f(x_k)\}$ and $\{W_k\}$ converge to the same limit;*

(continued)

Proposition 24.10 (continued)

$$(c_3) \lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0;$$

$$(c_4) \lim_{k \in K, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

Proof Reasoning as in the proof of Proposition 24.8, we have that assertions (c₁), (c₂) and (c₃) follow from Proposition 24.1. Moreover, reasoning by contradiction, if we assume that (c₄) is false, we can find an infinite subsequence with index set $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \rightarrow \infty} x_k = \bar{x},$$

and

$$\lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} \neq 0, \quad \text{where } \|\bar{d}\| = 1. \quad (24.48)$$

Now, if we suppose that there exists $\bar{k} \in K_1$ such that, for all $k \in K_1, k \geq \bar{k}$, Algorithm 24.4 terminates with $\alpha_k = 0$, we can repeat the same reasonings followed in the proof of Proposition 24.8 and we get a contradiction to (24.48).

Therefore there must exist an infinite subset $K_2 \subseteq K_1$ such that $\alpha_k \neq 0$ for $k \in K_2$. We can distinguish the different cases.

If the algorithm terminates at Step 4 for an infinite subsequence we have that condition (24.43) must be satisfied and hence we can repeat a reasoning similar to that followed in the case $\alpha_k = 0$, using the Mean Value Theorem and taking into account assertion (c₃), so that we contradict (24.48).

If the algorithm terminates (for an infinite subsequence) at step 5 or at step 8, we can use again (appropriately) the mean Value Theorem in correspondence to each pair of inequalities appearing in (24.44), (24.45) and (24.46), which bound the variations of f . Thus, we get again, in the limit of some subsequence converging to \bar{x} , $\nabla f(\bar{x})^T \bar{d} = 0$, which contradicts (24.48). \square

24.7 Watchdog Techniques

The nonmonotone line search rules considered in the preceding section typically yield acceptability conditions that are much less restrictive than the standard monotone rules. However, each point $x_k + \alpha_k d_k$ must remain in the level set \mathcal{L}_0 , and this may cause difficulties, occasionally, when the starting point is located at the

bottom of a steep sided valley. In this case the behaviour of the method may depend critically on the choice of the starting point and on the value of the memory M .

An alternative approach to the definition of nonmonotone line searches can be that of performing some iterations using a *relaxed* acceptability criterion, which may consist, for instance, in accepting the first trial step-length. After a prefixed number of tentative steps, if a significant reduction of the objective point has not been achieved, then the algorithm backtracks to the last accepted point and performs a standard line search. A technique based on this idea and called *watchdog technique* has been introduced in [46] for globalizing recursive quadratic programming methods in a way that, under appropriate assumptions, the Maratos effect (see Sect. 22.5) can be prevented. Thus, global convergence can be ensured while preserving a superlinear convergence rate.

Although the method has been introduced in the case of constrained problems, the technique can be extended also to unconstrained minimization and in this section we describe unconstrained algorithms for minimizing a continuously differentiable function $f : R^n \rightarrow R$, which can also be identified with some differentiable penalty function for a constrained problem.

We define computational schemes in which some “watchdog” rule is combined with nonmonotone line searches, in order to preserve, as much as possible, the good local properties of search directions, while guaranteeing global convergence. An algorithm based on this approach has been proposed for the first time in [125] for globalizing Newton’s method. Here we describe the basic features of a somewhat different scheme, introduced in [134], where a *nonmonotone watchdog* acceptability criterion is combined with a nonmonotone linesearch.

The algorithm can be described by specifying:

- a sequence of main iterations indexed by $k = 0, 1, \dots$, that yield the *accepted points* x_k ;
- for each k a finite set of *inner steps*, based on some local model, that produce the tentative points z_k^{j+1} for $j = 0, 1, \dots, t_k$ starting from $z_k^0 = x_k$, with $t_k \leq N - 1$, where $N \geq 1$ is a given number;
- a criterion (*watchdog criterion*) for accepting the current tentative point;
- a *non monotone* line search algorithm that computes a step-size α_k along a (suitable) descent direction d_k starting from x_k , whenever no tentative point has been accepted.

During the local steps convergence is not enforced and each local iteration is “observed” but not modified (except, possibly, in some extreme case).

An important feature is that the tentative points can be placed also outside the level set \mathcal{L}_0 , while the accepted points x_k remain in \mathcal{L}_0 . Global convergence towards stationary points in \mathcal{L}_0 is guaranteed through the line search, but the linesearch is performed only when all points produced during the local phase have been rejected.

We suppose that the gradient ∇f is continuous on R^n and that at each point produced by the algorithm the gradient vector is available. We also assume that

at each accepted point x_k we can evaluate a search direction d_k such that the assumptions stated in Proposition 24.4 are satisfied, that is, for every k we have:

$$\begin{aligned}\nabla f(x_k)^T d_k &\leq -c_1 \|\nabla f(x_k)\|^p, \quad p > 0, \\ \|d_k\|^q &\leq c_2 \|\nabla f(x_k)\|, \quad q > 0\end{aligned}\tag{24.49}$$

where $c_1, c_2 > 0$ and $pq > 1$.

The local steps are performed at each k , starting from $z_k^0 = x_k$ by computing the points

$$z_k^{j+1} = z_k^j + p_k^j, \quad j = 0, 1, \dots, t_k$$

where $t_k \leq N - 1$ and p_k^j is the search direction generated by some local model. In practice, as we will see, it could be convenient (but not necessary) to choose $p_k^0 = d_k$.

The point z_k^{j+1} , for some j , will be accepted and redefined as x_{k+1} , when a suitable *nonmonotone watchdog test* is satisfied. A possible definition of a watchdog test can be that of accepting the tentative point z_k^{j+1} if

$$f(z_k^{j+1}) \leq W_k - \max\{\sigma_1(\|\nabla f(x_k)\|), \sigma_2(\|z_k^{j+1} - x_k\|)\}\tag{24.50}$$

where

$$W_k = \max_{0 \leq j \leq \min(k, M)} \{f(x_{k-j})\}$$

is the reference value and σ_1, σ_2 are forcing functions.

We note that the forcing function σ_1 has the role of guaranteeing that, if for an infinite subsequence a point is accepted during the watchdog phase, then convergence towards stationary points can be established. The forcing function σ_2 (together with the assumptions on d_k) allow us to satisfy the assumptions of Proposition 24.1 and hence enforces the condition $\|x_{k+1} - x_k\| \rightarrow 0$ and the convergence of the function values.

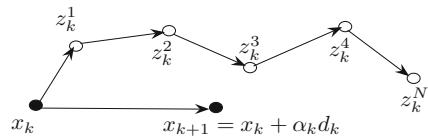
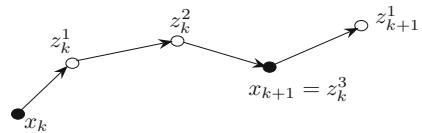
If all points z_k^j for $j = 1, \dots, N$ have been rejected, we backtrack to x_k and we perform a nonmonotone line search along d_k .

A schematic illustration of this strategy is given in Fig. 24.3. In the first example we suppose that the local algorithm produces the points z_k^1, z_k^2, z_k^3 . If z_k^3 satisfies the watchdog test we set $x_{k+1} = z_k^3$ and we start a new search from this point.

In the second example we suppose that all points generated locally do not satisfy the watchdog test and are rejected. In this case a linesearch is performed from x_k along d_k , by employing some nonmonotone line search technique and we set $x_{k+1} = x_k + \alpha_k d_k$, where α_k is the step-size.

On the basis of the preceding discussion, we can define the following (conceptual) algorithm model of a nonmonotone watchdog strategy.

Fig. 24.3 Two examples of the watchdog phase



Algorithm 24.5 (Nonmonotone Watchdog Algorithm)

Data: $x_0 \in R^n$, integers $N \geq 1$, $M \geq 0$, $k = 0$.

While $\nabla f(x_k) \neq 0$ **do**

 1. Set $z_k^0 = x_k$ and *linesearch*= true.

 2. **For** $j = 0, 1, N - 1$

 Determine the point $z_k^{j+1} = z_k^j + p_k^j$;
 if the *watchdog* test is satisfied, that is:

$$f(z_k^{j+1}) \leq W_k - \max\{\sigma_a(\|\nabla f(x_k)\|), \sigma_b(\|z_k^{j+1} - x_k\|)\}$$

 set $x_{k+1} = z_k^{j+1}$, *linesearch* = false and **exit** from step 2.

End For

 3. **If** *linesearch* = true **then**

 compute d_k such that Conditions (24.49) are satisfied;
 determine a step-size α_k along d_k using Algorithm 24.2;
 set $x_{k+1} = x_k + \alpha_k d_k$.

End if

 4. Set $k = k + 1$.

End While

Convergence of the algorithm is established in the following proposition.

Proposition 24.11 (Convergence of Algorithm 24.5) Suppose that the function $f : R^n \rightarrow R$ is continuously differentiable on R^n and that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence produced by Algorithm 24.5 and

(continued)

Proposition 24.11 (continued)

suppose that $\nabla f(x_k) \neq 0$ for all k . Then the sequence has limit points and every limit point is a stationary point of f in \mathcal{L}_0 . No limit point is a local unconstrained maximizer of f .

Proof When x_{k+1} is computed at step 2 during the watchdog phase, we have

$$f(x_{k+1}) \leq W_k - \sigma_2(\|x_{k+1} - x_k\|). \quad (24.51)$$

When x_{k+1} is obtained through the line search we can repeat the same reasonings in the proof of Proposition 24.6 and then conclude that there exists a forcing function σ such that

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|). \quad (24.52)$$

Thus the assumptions of Proposition 24.3 are satisfied and hence the assertions of this proposition must hold. In particular, we have that $\{x_k\}$ has limit points (by compactness of \mathcal{L}_0), that the sequences $\{W_k\}$ and $\{f(x_k)\}$ converge to the same limit and that $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$. Now, suppose first that there exists an infinite subsequence $\{x_k\}_{K_1}$ such that x_{k+1} is obtained through the line search algorithm. It easily verified that now all the assumptions of Proposition 24.6 are satisfied and hence we have

$$\lim_{\substack{k \in K_1, k \rightarrow \infty}} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

for every subsequence $\{x_k\}_{K_1}$ with $K_1 \subseteq K$, converging to some $\bar{x} \in \mathcal{L}_0$. On the other hand, by the assumption made on d_k we have

$$\frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \geq \frac{c_1 \|\nabla f(x_k)\|^p}{c_2^{1/q} \|\nabla f(x_k)\|^{1/q}} = (c_1/c_2^{1/q}) \|\nabla f(x_k)\|^{(qp-1)/q},$$

and hence, by continuity of ∇f , we get:

$$\nabla f(\bar{x}) = 0. \quad (24.53)$$

Suppose now that there exists an infinite subsequence, $\{x_k\}_{K_2}$ converging to a point $\hat{x} \in \mathcal{L}_0$ and such that for sufficiently large value of k the point x_{k+1} has been accepted at step 2. This implies, in particular)

$$f(x_{k+1}) \leq W_k - \sigma_a(\|\nabla f(x_k)\|),$$

and hence, taking limits for $k \rightarrow \infty$, $k \in K_2$ we have $\lim_{k \in K_2, k \rightarrow \infty} \sigma_a(\|\nabla f(x_k)\|) = 0$. We can conclude that every limit point of $\{x_k\}$ is a stationary point.

Finally it follows from Proposition 24.3 that no limit point can be a local maximizer of f . \square

Algorithm 24.5 can be employed in connection with different methods. Here we consider the application to Newton's method.

24.8 Nonmonotone Globalization of Newton's Method

We can specialize the watchdog algorithm introduced in the preceding section to globalization of Newton's method for unconstrained minimization. A conceptual model of this algorithm is given below.

Algorithm 24.6 (Nonmonotone Globalization of Newton Method)

Data. $x_0 \in R^n$, integer $N \geq 1$, $M \geq 0$, $k = 0$.

While $\nabla f(x_k) \neq 0$ **do**

1. Set $z_k^0 = x_k$ and *linesearch* = true.

2. **For** $j = 0, 1, N - 1$

if the system $\nabla^2 f(z_k^j)s = -\nabla f(z_k^j)$ has a solution,

determine a solution s^N and set $p_k^j = s^N$;

otherwise determine a search direction p_k^j and set $z_k^{j+1} = z_k^j + p_k^j$.

If the *watchdog test* is satisfied, that is, if:

$$f(z_k^{j+1}) \leq W_k - \max\{\sigma_a(\|\nabla f(x_k)\|), \sigma_b(\|z_k^{j+1} - x_k\|)\},$$

where σ_a, σ_b are forcing functions, then set $x_{k+1} = z_k^{j+1}$, *linesearch* = false and **exit** from step 2.

End For

3. **If** *linesearch* = true **then**

compute a search direction d_k such that:

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^2 \quad \|d_k\| \leq c_2 \|\nabla f(x_k)\|, \quad (24.54)$$

determine step-size α_k along d_k using non monotone Armijo's method, starting from a step-size $a = 1$ and set $x_{k+1} = x_k + \alpha_k d_k$.

End if

4. Set $k = k + 1$.

End While

We note that during the while cycle at Step 2, no specific assumption is made on the search direction p_k^j when the system

$$\nabla^2 f(z_k^j)s = -\nabla f(z_k^j)$$

has no solution. However, we can suppose that p_k^j is some modified Newton direction.

The forcing functions at the watchdog step can be defined in a way that convergence properties of Newton method are preserved and the algorithm is a *globally convergent modification of Newton's method* in the sense of Definition 13.1. In particular, at Step 2 we can consider the forcing functions

$$\sigma_a(t) = a_1 \min(1, t^3), \quad \sigma_b(t) = a_2 \min(1, t^3), \quad (24.55)$$

where a_1, a_2 are given positive numbers.

The search direction d_k computed at Step 2 can be some modified Newton direction and possibly the same direction p_k^0 used at Step 1. On the basis of the choice of d_k , we can also employ the linesearch Algorithm 24.2.

Proposition 24.12 (Convergence of Algorithm 24.6) *Let $f : R^n \rightarrow R$ be twice continuously differentiable on R^n and assume that the level set \mathcal{L}_0 is compact. Then Algorithm 24.6, with the choices (24.55), for the forcing functions, is a globally convergent Newton-type method. More specifically we have:*

- (i) *there exist limit points of $\{x_k\}$ and every limit point is a stationary point of f in \mathcal{L}_0 ;*
- (ii) *no limit point of $\{x_k\}$ is a local maximizer of f ;*
- (iii) *if $\{x_k\}$ converges towards a local minimum point x_\star of f and $\nabla^2 f$ is a positive definite matrix that satisfies the assumptions of Proposition 13.2, then there exists k^\star such that, for all $k \geq k^\star$ we have*

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

and hence the iteration of the algorithm is the pure Newton iteration.

Proof Recalling Remark 24.1, Proposition 24.11 implies that (i) and (ii) are satisfied. Let now \bar{x} be a limit point and relabel $\{x_k\}$ a subsequence converging to \bar{x} . Suppose that $\{x_k\}$ converges to a local minimizer x_\star such that $\nabla^2 f(x_\star)$ is positive definite and the assumptions of Proposition 13.2 are satisfied. First we show that, for sufficiently large values of k , the direction p_k^0 coincides with Newton's direction.

As ∇f^2 is continuous and $\nabla^2 f(x_*)$ is positive definite, we can find a neighborhood $B(x_*; \varepsilon_1)$ such that, for all $x \in B(x_*; \varepsilon_1)$, the matrix $\nabla^2 f(x)$ is positive definite and we have,

$$m\|z\|^2 \leq z^T [\nabla^2 f(x)]^{-1} z \leq M\|z\|^2, \quad \text{for all } z \in R^n$$

where $0 < m \leq M$ are suitable bounds on the eigenvalues of $[\nabla^2 f(x)]^{-1}$. Then, as $x_k \rightarrow x_*$, for sufficiently large values off k , say $k \geq k_1$ we can assume $x_k \in B(x_*; \varepsilon_1)$. Therefore we can compute Newton's direction

$$p_k^0 = s_k^N = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k).$$

and moreover, we can write

$$|\nabla f(x_k)^T p_k^0| = \nabla f(x_k)^T [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \geq m\|\nabla f(x_k)\|^2,$$

and

$$\|p_k^0\| = [\nabla f(x_k)[\nabla^2 f(x_k)]^{-2} \nabla f(x_k)]^{1/2} \leq M\|\nabla f(x_k)\|.$$

As $\nabla f(x_k) \rightarrow 0$, there exists $k_2 \geq k_1$ such that, for $k \geq k_2$ we have

$$\gamma |\nabla f(x_k)^T p_k^0| \geq \gamma m\|\nabla f(x_k)\|^2 \geq a_1\|\nabla f(x_k)\|^3, \quad (24.56)$$

where $\gamma < 1/2$ is a positive constant. On the basis of the preceding observations, recalling that $p_k^0 \rightarrow 0$, we can write, for large k :

$$\gamma \nabla f(x_k)^T p_k^0 \leq -\gamma m\|\nabla f(x_k)\|^2 \leq -\gamma \frac{m}{M^2} \|p_k^0\|^2 \leq -a_2 \|p_k^0\|^3. \quad (24.57)$$

Let now $B(x_*; \varepsilon) \subseteq B(x_*; \varepsilon_1)$ be the neighborhood considered in Proposition 13.2. Then, for large k , say $k \geq k_3 \geq k_2$, by Newton's convergence theory (recall Remark 13.2), we have necessarily that, for some $\beta > 0$:

$$\|x_k + p_k^0 - x_*\| \leq \beta \|x_k - x_*\|^2. \quad (24.58)$$

From Proposition 13.4, for large k (let $k \geq k^* \geq k_3$), we have for $\gamma < 1/2$

$$f(x_k + p_k^0) \leq f(x_k) + \gamma \nabla f(x_k)^T p_k^0 \leq W_k + \gamma \nabla f(x_k)^T p_k^0,$$

whence, taking into account (24.56) and (24.57), we get

$$f(x_k + p_k^0) \leq W_k - a_1\|\nabla f(x_k)\|^3,$$

$$f(x_k + p_k^0) \leq W_k - a_2 \|p_k^0\|^3.$$

Then, for $k \geq k^*$ the watchdog test is satisfied at z_k^1 , which implies $x_{k+1} = z_k^1 = x_k + p_k^0$, and the algorithm is a globally convergent Newton-type method. \square

24.9 Nonmonotone Inexact Newton-Type Methods for Nonlinear Equations

We consider the problem of solving the system

$$F(x) = 0, \quad (24.59)$$

where $F : R^n \rightarrow R^n$ is a continuously differentiable function, whose Jacobian matrix at x is denoted by $J(x)$.

Many works have been devoted to the study of nonmonotone globalization methods for nonlinear equations. Here we will restrict our attention to a nonmonotone strategy based on the combination of a nonmonotone Armijo-type derivative-free method with a nonmonotone watchdog rule, which can be viewed as an adaptation of the strategy introduced in the preceding sections.

We will refer to the case where the Jacobian matrix is not available and search directions are computed through an *inexact forward difference method* (more precisely, the FDGMRES method [136]). This technique requires the finite difference approximation of the product of $J(x)$ times a vector v by means of the difference quotient

$$[F(x + \sigma v) - F(x)]/\sigma, \quad (24.60)$$

with $\sigma > 0$. Under suitable assumptions, it has been shown that, for sufficiently small values of σ , and for some $\eta > 0$ the FDGMRES method can compute an *inexact Newton direction*, d satisfying the condition

$$\|J(x)d + F(x)\| \leq \eta\|F(x)\|, \quad (24.61)$$

where $\|\cdot\|$ is some given norm on R^n .

We suppose that a sequence $\{x_k\}$ is generated by means of an iteration of the form

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

where α_k is a step-size. We will show that the globalization algorithm defined in Chap. 16 can be modified, by introducing a reference value of the form

$$W_k = \max_{0 \leq j \leq \min(k, M)} \{ \|F(x_{k-j})\| \}, \quad (24.62)$$

and by combining nonmonotone watchdog rules with nonmonotone line searches.

The merit function employed is the (non differentiable) function

$$f(x) = \|F(x)\|$$

and the level set \mathcal{L}_0 is that corresponding to f and some given point $x_0 \in R^n$, that is:

$$\mathcal{L}_0 = \{x \in R^n : \|F(x)\| \leq \|F(x_0)\|\}.$$

As in Sect. 24.3, we will denote by $x_{\ell(k)}$, with $k - M \leq \ell(k) \leq k$, the point where the maximum defining W_k is attained. Thus, we have

$$f(x_{\ell(k)}) \equiv \|F(x_{\ell(k)})\| = W_k.$$

The next lemma follows immediately from Lemma 24.1.

Lemma 24.2 *Suppose that, for all k we have*

$$f(x_{k+1}) \equiv \|F(x_{k+1})\| \leq W_k. \quad (24.63)$$

Then:

- (i) *the sequences $\{W_k\}$ is monotonically non increasing;*
- (ii) *$x_k \in \mathcal{L}_0$ for all k .*

□

The next proposition is at the basis of the convergence results established in the sequel and takes into account the special structure of the problem.

Proposition 24.13 *Let $\{x_k\}$ be a sequence of points such that*

$$f(x_{k+1}) \leq W_k - \sigma(W_k), \quad (24.64)$$

where $\sigma : R^+ \rightarrow R^+$ is a forcing function. Then $x_k \in \mathcal{L}_0$ for all k and we have:

$$\lim_{k \rightarrow \infty} f(x_k) = \lim_{k \rightarrow \infty} W_k = 0.$$

Proof Condition (24.64) obviously implies that inequality (24.63) holds and hence it follows from Lemma 24.2 that $x_k \in \mathcal{L}_0$ for all k and that the sequence $\{f(x_{\ell(k)})\} \equiv \{W_k\}$ is non increasing. As $W_k \geq 0$, this implies that the sequence $\{W_k\}$ admits a limit $W_* \geq 0$ for $k \rightarrow \infty$. If $W_* = 0$, then inequality (24.64) implies that also $\{f(x_k)\}$ converges to zero and the assertion is proved. Therefore, reasoning by contradiction, we can assume that for sufficiently large k , say for $k \geq k^*$, we have

$W_k \geq t$ for some $t > 0$ and for all $k \geq k^*$. This in turn implies, by definition of forcing function, that there exists k_σ such that

$$\sigma(W_k) \geq t_\sigma, \quad \text{for some } t_\sigma > 0 \text{ and for all } k \geq k_\sigma.$$

It follows that, for all $k \geq k_\sigma$, condition (24.64) becomes

$$f(x_{k+1}) \leq W_k - t_\sigma. \quad (24.65)$$

On the other hand, letting $k \geq k_\sigma + M + 1$, so that $\ell(k) - 1 \geq k - M - 1 \geq k_\sigma$ and using (24.65), where k is replaced by $\ell(k) - 1$, we can write

$$f(x_{\ell(k)}) \leq f(x_{\ell(\ell(k)-1)}) - t_\sigma. \quad (24.66)$$

Therefore, taking limits and recalling that $\{f(x_{\ell(k)})\}$ converges to W_* , we have $t_\sigma = 0$, which contradicts our assumptions. \square

Choosing $\sigma(t) = \gamma t$, for $\gamma \in (0, 1)$, and letting $\tau = 1 - \gamma$ condition (24.64) can be satisfied by imposing

$$f(x_{k+1}) \leq \tau W_k, \quad (24.67)$$

where $\tau \in (0, 1)$.

We remark that a condition essentially equivalent to that stated above has been used in [106] within the convergence proof of the nonmonotone algorithm proposed there.

Before describing in detail the globalization algorithm, we first illustrate the scheme of a nonmonotone Armijo-type derivative-free linesearch method and the assumptions under which an approximate Newton direction can be computed.

A Nonmonotone Derivative-Free Linesearch

The scheme we consider here is an Armijo-type algorithm where, given a point x_k and a direction $d_k \in R^n$, starting from some fixed initial estimate $a \in (0, 1]$, we try to determine a non zero step-size $\alpha_k \in (0, 1]$ that satisfies the condition of sufficient reduction:

$$\|F(x_k + \alpha d_k)\| \leq (1 - \gamma \alpha) W_k, \quad (24.68)$$

where W_k is the reference value defined by (24.62) and $\gamma \in (0, 1)$. However, if the step-size is reduced below a prescribed bound μa , the search is terminated with a null step and the last value of the step-size that violates the acceptability condition is denoted by δ_k . A formal description is given in the following scheme, which will be referred to as subroutine NDFLS(x_k, d_k, μ, α_k).

Algorithm 24.7 (Nonmonotone Derivative-Free Line Search (NDFLS))

Parameters. $\gamma \in (0, 1)$, $a \in (0, 1]$, $0 < \xi_l < \xi_u < 1$.

Step 1. Set $\alpha = a$ and $j = 0$

Step 2. While $\|F(x_k + \alpha_k d_k)\| > (1 - \gamma\alpha)W_k$ **do**

If $\alpha < \mu a$ **then**

set $\alpha_k = 0$, $\delta_k = \alpha$ and **terminate**,

Else

choose $\xi \in [\xi_l, \xi_u]$

and set $\alpha = \xi\alpha$, $j = j + 1$.

End if

End while

Step 3. Set $\alpha_k = \alpha$ and **terminate**. □

We state the following proposition.

Proposition 24.14 Suppose that $F(x_k) \neq 0$ and let $\mu \in (0, 1)$ be given at x . Then Algorithm NDFLS(x_k, d_k, μ, α_k) determines, in a finite number of iterations, a scalar $\alpha_k \in [0, a]$ such that

$$\|F(x_k + \alpha_k d_k)\| \leq (1 - \gamma\alpha_k)W_k \quad (24.69)$$

and at least one of the following conditions holds:

$$\alpha_k = 0 \quad \text{and} \quad \|F(x_k + \delta_k d_k)\| > (1 - \gamma\delta_k)W_k \geq (1 - \gamma\delta_k)\|F(x_k)\| \quad \text{with } \delta_k < \mu a \quad (24.70)$$

$$\alpha_k \geq \xi_l \mu a. \quad (24.71)$$

Proof As the tentative step-size α is reduced by a factor $\xi \leq \xi_u < 1$ at each iteration, the algorithm terminates, either at Step 2 with $\alpha_k = 0$ or at Step 3 with $\alpha_k > 0$. In both cases we have obviously that α satisfies (24.69). In the first case, taking into account the instructions of Step 1 and Step 2, the fact that $\|F(x_k)\| \leq W_k$ and the inequality $\gamma\delta < \gamma\mu a < 1$, we have also that the scalar δ_k computed at Step 2 must satisfy condition (24.70). In the second case, termination occurs at Step 3 either with $\alpha_k = a$ or with $\alpha_k < a$. In the latter case, since the algorithm did not terminate at the previous step, this implies that $\alpha/\xi \geq \mu a$, and hence, as $\xi \geq \xi_l$, we have that (24.71) holds. □

When $d \in R^n$ is an inexact-Newton direction, Algorithm NDFLS may terminate, in general, with a zero step-size, which would correspond to a linesearch failure. However, we can state the following proposition, which can be easily derived from Lemma 8.2.1 in [152].

Proposition 24.15 Let $x \in R^n$ be any point such that $F(x) \neq 0$. Let $d \in R^n$ a vector satisfying

$$\|J(x)d + F(x)\| \leq \eta \|F(x)\|, \quad (24.72)$$

with $\eta \leq \bar{\eta} < (1 - \gamma)$ and $\gamma \in (0, 1)$. Assume that J is Lipschitz continuous, with Lipschitz constant L_J , on the closed ball

$$\bar{B}(x, r) = \{y \in R^n : \|y - x\| \leq r\}.$$

Suppose also that J is nonsingular on $\bar{B}(x, r)$ and let $m_J > 0$ be such that $\|J^{-1}(x)\| \leq m_J$, for all $y \in \bar{B}(x, r)$. Then, we have

$$\|F(x + \alpha d)\| \leq (1 - \gamma\alpha) \|F(x)\|,$$

for $\alpha \in [0, \alpha(x)]$, where

$$\alpha(x) = \min \left(\frac{r}{m_J(1 + \bar{\eta}) \|F(x)\|}, \frac{2(1 - \gamma - \bar{\eta})}{(1 + \bar{\eta})^2 m_J^2 L_J \|F(x)\|} \right).$$

□

If the assumptions of Proposition 24.15 are satisfied at x_k on some closed ball $\bar{B}(x_k, r)$, in correspondence to a Newton-type search direction d , it is easily seen that Algorithm NDFLS terminates finitely with a step-size $\alpha \in (0, \bar{\alpha}(x_k)]$ for sufficiently small values of σ . It will be shown in the sequel that this permits the construction of globally convergent Newton-type algorithms.

Computation of an Approximate Newton Direction

Under appropriate assumptions, it has been shown that the FDGMRES method can compute an approximate Newton-type direction for sufficiently small values of the finite-difference step σ .

More precisely the following result holds (see, in particular, Theorem 3.6 of [34] and Proposition 6.2.1 of [152])

Proposition 24.16 Given $x_k \in R^n$, suppose that:

- (i) there exists a convex set Ω_k such that $x_k \in \Omega_k$, where J is non singular and Lipschitz continuous with constant L_{J_k} ;

(continued)

Proposition 24.16 (continued)
(ii) *there exists c_k such that*

$$\|J(y)^{-1}\| \leq c_k, \quad (24.73)$$

for all $y \in \Omega_k$.

Let

$$\hat{\sigma}_k = \frac{1}{2n^{1/2}L_{J_k}c_k} \quad (24.74)$$

and

$$C_k = 4n^{1/2}L_{J_k}c_k. \quad (24.75)$$

Then, for any $\sigma \in (0, \hat{\sigma}_k]$, and for any $\eta \in (0, 1)$, procedure FDGMRES determines a direction d_k satisfying

$$\|J(x_k)d_k + F(x_k)\| \leq (\eta + C_k\sigma) \|F(x_k)\|. \quad (24.76)$$

□

We indicate by $\text{FDGMRES}(z, \sigma, \eta, p)$ the routine that computes at the point z an approximate Newton direction p , for given values of σ and η .

A Nonmonotone Globalization Algorithm

The (conceptual) algorithm model defined here is essentially a (more detailed) nonmonotone version of the *Inexact Newton's Algorithm* defined in Chap. 16. The algorithm is based on the combination of a nonmonotone “watchdog” phase with nonmonotone linesearches along the approximate Newton directions.

At any major iteration k , the watchdog phase is performed, starting from the current point $x_k = z^0$, by computing, for $i \in \{0, \dots, N - 1\}$ the tentative points

$$z^{i+1} = z^i + p^i,$$

where p^i is the approximate Newton step obtained by employing the procedure $\text{FDGMRES}(z^i, \sigma, \eta, p^i)$.

In this phase, both the finite-difference step-size σ and the forcing parameter η are given and, in general, may also depend on k , according to suitable criteria.

A tentative point is accepted whenever an acceptability condition (“nonmonotone watchdog test”) of the form

$$\|F(z^{i+1})\| \leq \rho W_k, \quad \rho \in (0, 1)$$

is satisfied.

If the test is satisfied, then we assume $x_{k+1} = z^{i+1}$ as the new iterate and a new main iteration is started.

When no tentative point is accepted during the prefixed number of N steps, we backtrack to x_k and we attempt initially to perform a nonmonotone linesearch using Algorithm NDFLS along the search direction p^0 .

We suppose that in Algorithm NDFLS the initial tentative step-size is

$$a = 1.$$

As the assumptions of Proposition 24.15 may be not satisfied in a neighborhood of the current point, there is no guarantee that the linesearch algorithm can compute a non zero step-size. Therefore, in case of failure, the search direction p^0 is recomputed by means of Algorithm FDGMRES(x_k, σ, η, p^0), for decreasing values of the finite-difference step-size σ and of the forcing parameter η and a new linesearch is carried out with a smaller termination tolerance μ .

Assuming exact arithmetic, the results of Proposition 24.15 and Proposition 24.16 guarantee, under appropriate assumptions, that this process must terminate finitely with a new updated point x_{k+1} .

In this case we denote (in the proof), respectively, by $\tilde{\sigma}_k, \tilde{\eta}_k, \tilde{\mu}_k$ the values of finite-difference step-size, the forcing parameter and the termination tolerance, obtained at the end of the linesearch phase.

In the model algorithm reported below we assume that the termination criterion is given by

$$F(x) = 0$$

and we omit many practical details.

Algorithm 24.8 (Nonmonotone Globalization Algorithm)

Parameters $x_0 \in R^n, N \geq 1, M \geq 0, \gamma \in (0, 1), \bar{\eta} \in (0, 1 - \gamma), \bar{\mu} \in (0, 1), \rho \in (0, 1), \theta_1, \theta_2, \theta_3 \in (0, 1)$, and sequences $\{\sigma_k\}, \{\eta_k\}$ such that $0 < \sigma_k, 0 < \eta_k \leq \bar{\eta}$,

Step 0. Set *watchdog*=true and $k = 0$

While the stopping criterion is not satisfied **do**

(continued)

Algorithm 24.8 (continued)

```

Step 1. Set linesearch=true.
Step 2. If watchdog=true then
    Set  $z^0 = x_k$ ,  $\sigma = \sigma_k$ ,  $\eta = \eta_k$ ,  $\mu = \bar{\mu}$  and  $t = 1$ 
    For  $i = 0, N - 1$ 
        Compute a direction  $p^i$  using FDGMRES( $z^i, \sigma, \eta, p^i$ ),
        and set  $z^{i+1} = z^i + p^i$ 
        If  $\|F(z^{i+1})\| \leq \rho W_k$  then
            set  $x_{k+1} = z^{i+1}$ , linesearch=false,  $k = k + 1$  and exit
            from Step 2
        End If
    End For
    End If

Step 3. If linesearch=true then
    Compute  $\alpha$  by means of Algorithm
    NDPLS( $x_k, p^0, \mu, \alpha$ )
    If  $\alpha = 0$  then
        set  $\sigma = \theta_1 \sigma$ ,  $\eta = \theta_2 \eta$ , compute again  $p^0$  using
        FDGMRES( $x_k, \sigma, \eta, p^0$ ),
        set  $\mu = \theta_3 \mu$ ,  $t = t + 1$  and watchdog=false.
    Else
        set  $\alpha_k = \alpha$ ,  $d_k = p^0$ 
        set  $\tilde{\sigma}_k = \sigma$ ,  $\tilde{\eta}_k = \eta$ ,  $\tilde{\mu}_k = \mu$ 
        set  $x_{k+1} = x_k + \alpha_k d_k$ , watchdog=true, and  $k = k + 1$ 
    End If
End If

End While

```

The convergence of the algorithm is established in the next proposition .

Proposition 24.17 *Let $\{x_k\}$ be the sequence generated by Algorithm 24.8. Assume that there exists $r > 0$ such that for every $x \in \mathcal{L}_0$ the closed ball $\bar{B}(x, r)$ is contained in a open convex set Ω , where J is non singular and Lipschitz continuous, with Lipschitz constant L_J . Suppose also there exists $m_J > 0$ such that $\|J^{-1}(x)\| \leq m_J$ for all $x \in \Omega$.*

Then Algorithm 24.8 is well defined and either terminates at some x_p such that $F(x_p) = 0$ or it generates an infinite sequence $\{x_k\}$ such that

$$\lim_{k \rightarrow \infty} F(x_k) = 0. \quad (24.77)$$

Proof The instructions of the algorithm and the definition of W_k imply that if a new point x_{k+1} is generated, starting from some $x_k \in \mathcal{L}_0$, then we have necessarily that

$$\|F(x_{k+1})\| \leq W_k = \max_{0 \leq j \leq \min(k, M)} \{\|F(x_{k-j})\|\}. \quad (24.78)$$

Therefore, reasoning as in the proof of Lemma 24.2, we have that

$$W_j \leq W_{j-1}, \quad j = 1, \dots, k$$

As $W_0 = \|F(x_0)\|$, by induction, it follows that, if the points x_0, \dots, x_k have been computed by the algorithm, we have necessarily that

$$\|F(x_j)\| \leq \|F(x_0)\|, \quad \text{for all } j = 0, 1, \dots, k \quad (24.79)$$

so that the points x_j , for $j = 0, 1, \dots, k$ belong to the level set \mathcal{L}_0 .

In order to establish the thesis, first we must show that the algorithm cannot get stuck at some point x_k , where $F(x_k) \neq 0$, by repeating infinitely Step 3, without generating a new point x_{k+1} .

Reasoning by contradiction, let us assume that for some $x_k \in \mathcal{L}_0$ we have $\alpha = 0$ each time that Algorithm NDFLS(x_k, p^0, μ, α) is called. Let t be the iteration index, defined at Step 3, that counts the number of calls of Algorithm NDFLS at iteration k , let $\sigma(t), \eta(t)$ be the corresponding input parameters of procedure FDGMRES, and let $\mu(t)$ be the termination tolerance used in Algorithm NDFLS, so that $t \rightarrow \infty$, and we have $\sigma(t) \rightarrow 0, \eta(t) \rightarrow 0$ and $\mu(t) \rightarrow 0$ for $t \rightarrow \infty$. As $x_k \in \mathcal{L}_0 \subseteq \Omega$, it follows from Proposition 24.16 that, for $\sigma(t)$ and $\eta(t)$ sufficiently small, that is for $\sigma(t) \leq 1/(2n^{1/2}L_J m_J)$, and $\eta(t) + C\sigma(t) < (1 - \gamma)$, procedure FDGMRES($x_k, \sigma(t), \eta(t), p^0$) determines a direction p^0 such that

$$\|J(x_k)p^0 + F(x_k)\| \leq (\eta(t) + C\sigma(t))\|F(x_k)\| < (1 - \gamma)\|F(x_k)\|, \quad (24.80)$$

with $C = 4n^{1/2}L_J m_J$. On the other hand, as the closed ball $\bar{B}(x_k, r)$ is contained in Ω , it is easily verified that the assumptions of Proposition 24.15 are satisfied on $\bar{B}(x_k, r)$ with $d = p^0$ and we can consider the interval $(0, \bar{\alpha}(x_k)]$ defined there. From the instructions of Algorithm NDFLS it follows that for sufficiently large values of the iteration index j we have that $\alpha(j) \leq \bar{\alpha}(x_k)$, where $\alpha(j)$ is the tentative step-size at step j . As $a = 1$ and $\xi \in [\xi_l, \xi_u]$, we have at step j that $\xi_l^j \leq \alpha(j) \leq \xi_u^j$, so that we can choose an integer j^* satisfying

$$j^* \geq \max \left\{ 0, \frac{\log(\bar{\alpha}(x_k))}{\log(\xi_u)} \right\},$$

for which $\alpha(j^*) \leq \bar{\alpha}(x_k)$. On the other hand, as $\mu(t) \rightarrow 0$, we can assume that t is sufficiently large to have that $\mu(t) < \xi_l^{j^*} \leq \alpha(j^*)$, so that Algorithm NDFLS must terminate with a positive stepsize α_k and this contradicts the assumption that

the point x_{k+1} is never generated at x_k . It can be concluded that the algorithm is well defined and either terminates finitely at some point x_k such that $F(x_k) = 0$, or it produces an infinite sequence such that either

$$\|F(x_{k+1})\| \leq \rho W_k \quad (24.81)$$

when x_{k+1} is obtained at Step 2 during the watchdog phase, or else

$$\|F(x_{k+1})\| \leq (1 - \gamma \alpha_k) W_k, \quad (24.82)$$

when x_{k+1} is computed at Step 3 through a linesearch.

Now, we show that, if a stepsize $0 < \alpha_k \leq 1$ is computed at Step 3 for an infinite subsequence, say for $k \in K$, then α_k is bounded away from zero for $k \in K$, that is that there exists $\tilde{\alpha}$ such that

$$\alpha_k \geq \tilde{\alpha}, \quad \text{for all } k \in K. \quad (24.83)$$

Reasoning by contradiction, suppose that (24.83) is false and hence that there exists a subsequence $\{x_k\}_{K_1}$ with $K_1 \subseteq K$ such that $\alpha_k \rightarrow 0$ for $k \in K_1, k \rightarrow \infty$. Because of the instructions of Algorithm NDFLS, this is possible only if μ has been reduced for an infinite subsequence and this implies, in turn, that we have also $\tilde{\sigma}_k \rightarrow 0$ and $\tilde{\eta}_k \rightarrow 0$ for $k \in K_1, k \rightarrow \infty$. Recalling again Proposition 24.16, we have that for $\tilde{\sigma}_k \leq 1/(2n^{1/2}L_J m_J)$, and $\tilde{\eta}_k + C\tilde{\sigma}_k \leq \bar{\eta}$ procedure FDGMRES($x_k, \tilde{\sigma}_k, \tilde{\eta}_k, d_k$) determines a direction d_k such that

$$\|J(x_k)d_k + F(x_k)\| \leq (\tilde{\eta}_k + C\tilde{\sigma}_k)\|F(x_k)\| \leq \bar{\eta}\|F(x_k)\|, \quad (24.84)$$

where $\bar{\eta} < (1 - \gamma)$. As $\bar{B}(x_k, r) \subset \Omega$, the assumptions of Proposition 24.15 are satisfied and therefore by this proposition we have that

$$\|F(x_k + \alpha d_k)\| \leq (1 - \gamma \alpha)\|F(x_k)\| \leq (1 - \gamma \alpha)W_k, \quad \text{for all } \alpha \in [0, \bar{\alpha}(x_k)]. \quad (24.85)$$

As $x_k \in \mathcal{L}_0$ for all k , so that $\|F(x_k)\| \leq \|F(x_0)\|$, it follows from the expression of $\bar{\alpha}(x_k)$ that

$$\bar{\alpha}(x_k) \geq \bar{\alpha}(x_0) \quad \text{for all } k \in K_1. \quad (24.86)$$

On the other hand, as $\alpha_k \rightarrow 0$ for $k \in K_1$, for sufficiently large values of $k \in K_1$ we have necessarily that $\alpha_k < 1$ and hence, recalling the instructions of Algorithm NDFLS, we can assert that the stepsize

$$\frac{\alpha_k}{\xi_k} \leq \frac{\alpha_k}{\xi_l}$$

must violate the acceptability condition of Algorithm NDFLS. By (24.85) and (24.86), this implies that

$$\alpha_k \geq \min\{1, \xi_l \bar{\alpha}(x_k)\} \geq \min\{1, \xi_l \bar{\alpha}(x_0)\},$$

but this yields a contradiction to the assumption that $\alpha_k \rightarrow 0$ for $k \in K_1, k \rightarrow \infty$. It can be concluded that (24.83) holds for some $\bar{\alpha} \in (0, 1]$ and all $k \in K$ and therefore, recalling (24.81), for some $\tau \in (0, 1)$ we have, for all k :

$$\|F(x_{k+1})\| \leq \tau W_k,$$

where $\tau = \rho \in (0, 1)$ if $K = \emptyset$ or $\tau = \max\{\rho, \bar{\alpha}\} \in (0, 1)$ is $K \neq \emptyset$. Thus the assertion follows from Proposition 24.1 and the assumptions made. \square

The preceding result, stated under the assumption that the Jacobian matrix is nonsingular over a suitable region, obviously implies that every limit point of $\{x_k\}$ is a solution of the system. This yields also constructive assumptions for the existence of solutions of the system $F(x) = 0$. In the absence of the non singularity assumption on the Jacobian, it could be only possible to ensure the convergence towards stationary points of the merit function. In this case, the computation of a solution of the nonlinear system (provided that it exists) would require the adoption of global optimization techniques.

If we assume that σ_k and η_k become sufficiently small for $k \rightarrow \infty$, we have that Algorithm 24.8 will ultimately accept the first inexact Newton step in the watchdog phase and the convergence results established in the monotone case will be still valid in our case .

24.10 Notes and References

As already remarked, nonmonotone methods have been considered since many years, although the works on this subject have not been much popular for much time, at least in the western literature on optimization. Relevant exceptions has been, for instance, some non monotone algorithms proposed in the field of nonsmooth optimization (e.g. [240]), the *watchdog technique* of [46], the step-length acceptability criteria in [20] and methods related to Lyapunov-type conditions for the study of difference equations. In particular, the connections between the convergence theory of algorithms and the stability analysis of dynamical discrete-time systems have been considered in several works. We mention only the paper [199] on general iterative methods and the books [208] and [84], where explicit reference is made to optimization methods and to nonmonotone techniques.

In the present chapter we have confined our attention to some specific choices of a (Lyapunov-type) merit function. As already remarked, the basic works of our analysis has been the papers [122] and [125], in the context of globalization

strategies for Newton-type methods. Extensions of these works have been given to globalization of Gauss-Newton methods for nonlinear equations [90], to algorithms for nonlinear complementarity problems [80], to globalization of SQP methods [30, 201] and to globalization of the Barzilai-Borwein gradient method (see Chap. 25).

In this paper we have given a somewhat different formulation based on the introduction of a nonmonotone watchdog rule, derived from [134] and here extended to Newton-type methods. Applications of the same approach has been given to finite-difference Newton-type methods for nonlinear equations [136], to derivative-free Barzilai-Borwein method for nonlinear equations studied in Chap. 25 and to other derivative-free techniques [131].

In recent years the literature on nonmonotone optimization algorithms has largely expanded and we cannot give here a significant account of the various proposals. Here we only mention a few works with alternative formulations of the merit function [137, 249, 263], some papers on the extension of nonmonotone methods to trust region techniques [65, 250], works on derivative-free algorithms [104, 107], applications to interior point algorithms [28, 108], and a work on nonmonotone decomposition [102].

Chapter 25

Spectral Gradient Methods



In this chapter we report some basic results on a class of gradient methods for minimizing smooth functions, known as *spectral gradient methods*, which have been originated from the *Barzilai-Borwein (BB) gradient method* proposed in the seminal paper [11]. In the sequel we first illustrate the motivation and the basic features of the BB method in the quadratic case. Then we consider extensions to the minimization of non quadratic functions and to the solution of nonlinear equations. In particular, we describe nonmonotone globalization algorithms employing the nonmonotone line searches and the watchdog rules considered in Chap. 24. Finally, we report some results obtained in the literature in the case of constrained minimization problems.

25.1 The BB Method in the Quadratic Case

Let us consider the problem of minimizing the strictly convex quadratic function

$$f(x) = \frac{1}{2}x^T Qx - c^T x, \quad (25.1)$$

where Q is symmetric positive definite. The gradient of f at x is given by

$$\nabla f(x) = Qx - c$$

and the global minimizer is the point $x^* = Q^{-1}c$.

We already know that this problem can be solved through the steepest descent method, starting from any given initial point $x_0 \in R^n$, by employing the iterative algorithm

$$x_{k+1} = x_k - \alpha_k^* \nabla f(x_k),$$

where α_k^* is the “optimal” step-size, that is, the step-size that minimizes the objective function $f(x_k - \alpha \nabla f(x_k))$ with respect to α .

As shown in Chap. 11, we have

$$\alpha_k^* = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T Q \nabla f(x_k)}. \quad (25.2)$$

The steepest descent method converges with Q-linear convergence rate, but, in general, the method is much less efficient than other first order methods such as the conjugate gradient method or the Quasi-Newton methods.

In particular, the behaviour of the method depends critically on the condition number of Q and becomes quite inefficient in the presence of ill-conditioning, that is for large values of the ratio λ_M/λ_m , where λ_M, λ_m are the largest and the smallest eigenvalues of Q . On the other hand, we know (see Chap. 11) that the gradient method terminates finitely in at most n steps if the step-sizes along the negative gradients are chosen, in sequence, as the inverse eigenvalues of Q . In particular, if the gradient direction would coincide with an eigenvector of Q we could terminate in a single step. In practice, eigenvalues of Q are not easily available, but the preceding observations indicate that the inefficiency of the steepest descent method may depend also on the choice of the step-size along $-\nabla f$.

A new version of the gradient method, known as *Barzilai-Borwein gradient method* [11] consists in the definition of the step-size along the negative gradient direction, on the basis of information relative to the preceding step, in order to get some approximation of the Quasi-Newton equation. The Barzilai-Borwein gradient method is defined by the iteration

$$x_{k+1} = x_k - \frac{1}{\mu_k} \nabla f(x_k),$$

where μ_k is defined in a way that the Hessian Q is approximated with the matrix

$$B = \mu_k I,$$

or, equivalently, the inverse matrix Q^{-1} is approximated with a matrix of the form

$$H = (1/\mu_k)I,$$

where I is the identity matrix $n \times n$.

The approximation of Q (or of Q^{-1}) is constructed by computing the value of μ (or of $1/\mu$) that minimizes the error on the Quasi-Newton equation, which can be put in the form

$$Qs = y, \quad \text{or} \quad s = Q^{-1}y,$$

where we set, for $k > 0$:

$$s \equiv s_{k-1} = x_k - x_{k-1}, \quad y \equiv y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}).$$

(When convenient, to simplify notation, we omit the subscript $k - 1$ in y and s).

Then, the value of μ that minimizes the error on the Quasi-Newton equation $Bs - y = 0$ when $B = \mu I$ can be obtained by minimizing the function

$$\psi(\mu) = \|\mu s - y\|^2 = \mu^2 \|s\|^2 - 2\mu s^T y + \|y\|^2$$

and we obtain

$$\mu_k^a = \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}. \quad (25.3)$$

Similarly, the value of β that minimizes the error on the Quasi-Newton equation $s - Hy = 0$ when $H = \beta I$, is given by $\beta = 1/\mu_k^b$, with

$$\mu_k^b = \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}}. \quad (25.4)$$

When the objective function is a strictly convex quadratic function, the BB method can be viewed as a gradient method where the step-size is an approximation of the inverse of some eigenvalue of the Hessian matrix. For this reason the BB method is often referred to as the *spectral gradient method*.

In order to clarify this point, we recall that the eigenvalues can be approximated by evaluating the so-called *Rayleigh quotients*, defined by

$$R_Q(x) = \frac{x^T Q x}{\|x\|^2},$$

for every $x \in R^n$ with $x \neq 0$. It can be easily verified that, by assuming x as an eigenvector of Q , the corresponding eigenvalue is the Rayleigh quotient $R_Q(x)$. For x in R^n , the ratio $R_Q(x)$ remains (for real symmetric matrices) in the interval (known as *the numerical image* of Q) defined by $[\lambda_m(Q), \lambda_M(Q)]$ where $\lambda_m(Q), \lambda_M(Q)$ are the smallest and the largest eigenvalue of Q , that is

$$\lambda_m(Q) = \min_x R_Q(x), \quad \lambda_M(Q) = \max_x R_Q(x),$$

so that

$$\lambda_m(Q) \leq \frac{x^T Q x}{\|x\|^2} \leq \lambda_M(Q). \quad (25.5)$$

It can be easily verified that the values of μ^a, μ^b defined above correspond to particular Rayleigh quotients that can be evaluated on the basis of the preceding iteration, without requiring explicitly the knowledge of the matrix Q . In fact, as $y = Qs$, we have

$$\mu^a = \frac{s^T y}{\|s\|^2} = \frac{s^T Qs}{\|s\|^2} = R_Q(s).$$

Similarly, we can write

$$\mu^b = \frac{y^T y}{s^T y} = \frac{s^T Q Q s}{s^T Q s} = \frac{(Q^{1/2}s)^T Q (Q^{1/2}s)}{\|Q^{1/2}s\|^2} = R_Q(Q^{1/2}s).$$

Therefore, we can expect that μ^a and μ^b can yield approximations of the eigenvalues of Q , at least when the numerical image of Q is not too large.

In the sequel, we will refer to $\mu \equiv \mu^a$, but the extension to μ^b can be easily carried out. In order to simplify notation, we set

$$g_k = g(x_k) = \nabla f(x_k), \quad \alpha_k = 1/\mu_k.$$

An iteration of the BB method can put into the form

$$x_{k+1} = x_k - \alpha_k g_k,$$

which implies

$$g_{k+1} = g_k - \alpha_k Q g_k.$$

We have

$$\mu_k = \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} = \frac{g_{k-1}^T Q g_{k-1}}{\|g_{k-1}\|^2}.$$

whence it follows

$$\alpha_k = \frac{\|g_{k-1}\|^2}{g_{k-1}^T Q g_{k-1}}. \quad (25.6)$$

Thus, we can note that the BB step-size α_k at the k -th iteration is the “optimal” step-size at the preceding iteration, that is $\alpha_k = \alpha_{k-1}^*$.

As μ_k is a particular Rayleigh quotient, we have, for all k

$$0 < \lambda_{\min}(Q) \leq \mu_k \leq \lambda_{\max}(Q), \quad (25.7)$$

and hence $\alpha_k = 1/\mu_k$ is well-defined. Thus, we can define the following conceptual model.

BB Method in the Quadratic Case

Data: starting point $x_0 \in R^n$; initial step-size $\alpha_0 > 0$.

Set $g_0 = Qx_0 - c$, $k = 0$

While $g_k \neq 0$

 Set

$$\begin{aligned} x_{k+1} &= x_k - \alpha_k g_k, \\ g_{k+1} &= g_k - \alpha_k Qg_k, \\ \alpha_{k+1} &= \|g_k\|^2 / g_k^T Qg_k, \end{aligned}$$

 Set $k = k + 1$.

End While

It can be shown that the algorithm converges to the optimal solution. More specifically, the following result has been established [223].

Proposition 25.1 (Convergence of the BB Method in the Quadratic Case)

Let $f : R^n \rightarrow R$ be a strictly convex quadratic function and assume that the BB method generates a sequence $\{x_k\}$ such that $g_k \neq 0$ for all k . Then the sequence converges to the global minimum point x^* of f . \square

The convergence proof can easily be extended to the case when we refer to μ^b and also to the case when f is a convex quadratic function that admits a global minimizer in R^n , that is when Q is positive semidefinite and there exists x such that $b = Qx$.

In the work of Barzilai and Borwein it is shown that the method has R-superlinear convergence rate for two-dimensional convex quadratic functions. In the general n -dimensional case, it can be shown that the convergence rate is at least R-linear, that is, there exist constants c_a and c_b , $c_a > 0$, $0 < c_b < 1$, such that

$$\|g_k\| \leq c_a c_b^k.$$

In practice, the BB method is much more efficient than the steepest descent method and a possible explanation could be the fact that in many cases the vector g_k approximates, for $k \rightarrow \infty$, an eigenvector of Q , so that the global minimizer can be well approximated by a step along $-g_k$.

Many different modifications and criteria for the choice of the step-size have been proposed and the interested reader is referred to the literature.

25.2 Nonmonotone Globalization of the BB Method

In the general case, the BB method can be viewed as a gradient method where the search direction is defined by $d_k = -(1/\mu_k)\nabla f(x_k)$, where μ_k is computed, when possible, through the BB formulae.

When f is twice continuously differentiable we can write

$$\nabla f(x_k) = \nabla f(x_{k-1}) + \int_0^1 \nabla^2 f(x_{k-1} + t(x_k - x_{k-1}))(x_k - x_{k-1}) dt,$$

so that

$$\begin{aligned}\mu_k^a &= \frac{(\nabla f(x_k) - \nabla f(x_{k-1}))^T (x_k - x_{k-1})}{\|x_k - x_{k-1}\|^2} \\ &= \int_0^1 \frac{(x_k - x_{k-1})^T \nabla^2 f(x_{k-1} + t(x_k - x_{k-1}))(x_k - x_{k-1})}{\|x_k - x_{k-1}\|^2} dt.\end{aligned}$$

Thus μ_k^a can be viewed as a mean of the Rayleigh quotients $R(s_{k-1})$, when the Hessian matrix is evaluated along the segment with endpoints x_{k-1}, x_k .

However, if f is not a quadratic convex function, the numbers μ_k^a and μ_k^b given by the BB formulae could be not well defined and could also yield unacceptable values of the step-size μ_k . Therefore, we must modify, if needed, the values given by the BB formulae, in order that, for some small $\varepsilon > 0$, we have

$$\varepsilon \leq \mu_k \leq 1/\varepsilon. \quad (25.8)$$

Moreover, in order to enforce global convergence, we must perform a suitable line search along d_k . We can observe that in the quadratic convex case the value of μ_k does not imply a monotonic reduction of the function f and hence it would appear that a nonmonotone line search should be important in order to preserve, as much as possible, the convergence properties of the method.

A nonmonotone globalization technique (known as *Global Barzilai-Borwein method* [224] (GBB)) can be defined with the iteration

$$x_{k+1} = x_k - \eta_k \frac{1}{\mu_k} \nabla f(x_k)$$

where $0 < \eta_k \leq 1$ is computed by means of a nonmonotone Armijo-type algorithm, based on the acceptance condition

$$f(x_k + \eta d_k) \leq W_k + \gamma \eta \nabla f(x_k)^T d_k, \quad (25.9)$$

where $d_k = -(1/\mu_k) \nabla f(x_k)$, μ_k is the BB step-size (possibly modified for satisfying (25.8)) and W_k is the reference value introduced in Chap. 24, defined by

$$W_k = \max_{0 \leq j \leq \min(k, M)} \{f(x_{k-j})\},$$

with $M \geq 0$. Computational experience has indicated that the GBB method is much more efficient than the steepest descent method with monotone line search and it is competitive with the conjugate gradient method even in large dimensional convex problems. However, in many difficult highly nonlinear problems it could be convenient to permit a further relaxation of monotonicity, by adopting the scheme already introduced in Chap. 24, where a nonmonotone watchdog rule is employed in combination with a nonmonotone line search.

For $k = 0$ we suppose that μ_0 is a positive number, computed, for instance, through a linesearch along $-\nabla f(x_0)$. In the local iterations, defined by

$$z_k^{i+1} = z_k^i + p_k^i, \quad p_k^i = -\frac{1}{\mu_k^i} \nabla f(z_k^i)$$

for $i = 0, 1, \dots, N - 1$, the scalars μ_k^i are computed using, when possible, the BB formulae. When a linesearch is required, we set $d_k = p_k^0$. In principle, we can use different formulae for each i and, in particular, it could be convenient alternating the values μ^a and μ^b defined earlier.

Then we can refer to the scheme already introduced in Sect. 24.7 of Chap. 24, with suitable modifications, as indicated in the following conceptual model.

Algorithm 25.1 Data: $x_0 \in R^n$, integers $N \geq 1$, $M \geq 0$, $k = 0$.

While $\nabla f(x_k) \neq 0$ **do**

1. Set $z_k^0 = x_k$ and *linesearch*= true.
2. **For** $i = 0, 1, N - 1$
 - Determine $\mu_k^i \neq 0$ and compute the point

$$z_k^{i+1} = z_k^i - \frac{1}{\mu_k^i} \nabla f(z_k^i);$$

if the *watchdog* test is satisfied:

(continued)

Algorithm 25.1 (continued)

$$f(z_k^{i+1}) \leq W_k - \max\{\sigma_a(\|\nabla f(x_k)\|), \sigma_b(\|z_k^{i+1} - x_k\|)\}$$

set $x_{k+1} = z_k^{i+1}$, *linesearch*=false and **exit** from Step 2.

End For

3. If *linesearch*=true **then**

compute a step-size α_k along $d_k = -(1/\mu_k^0) \nabla f(x_k)$ by means
of

Algorithm 24.2 and set $x_{k+1} = x_k + \alpha_k d_k$.

End if

4. Set $k = k + 1$.

End While

The convergence properties of Algorithm 25.1 follow immediately from Proposition 25.2, taking into account the fact that all search directions (and hence d_k) satisfy conditions (24.49), because of assumption (25.8).

25.3 Spectral Gradient Methods for Nonlinear Equations

Under appropriate assumptions, spectral gradient methods can be employed for solving a system of nonlinear equations

$$F(x) = 0,$$

where $F : R^n \rightarrow R^n$ is a vector of continuously differentiable functions.

We consider, in particular, Jacobian-free methods employing at each iteration k the residual vector $-F(x_k)$ as search direction, so that we have:

$$x_{k+1} = x_k - \alpha_k F(x_k),$$

where α_k is a step-size. More specifically, we refer to the technique proposed in [160], where the BB method is employed for computing a scaling factor of the residual vector, but we consider a different algorithm.

First we introduce a globalization strategy and we state sufficient conditions for convergence. We make use of the merit function $f : R^n \rightarrow R$ defined by

$$f(x) = \frac{1}{2} \|F(x)\|^2,$$

where $\|\cdot\|$ is the Euclidean norm on R^n . The globalization method is based on the combination of the nonmonotone derivative-free line search Algorithm 24.4 introduced in Chap. 24 with a nonmonotone watchdog rule.

At any major iteration k the watchdog phase consists in a finite sequence of tentative points, starting from $z_k^0 = x_k$, which are generated through the iteration

$$z_k^{j+1} = z_k^j - \frac{1}{\mu_k^j} F(z_k^j), \quad j = 0, 1, \dots, t_k$$

where $t_k \leq N - 1$ and μ_k^j is a scaling factor, such that a condition of the form

$$l \leq |\mu_k^j| \leq u \quad (25.10)$$

is satisfied for all j and k , given $u > l > 0$.

We will choose the parameters μ_k^j as safeguarded estimates of BB step-sizes; however, we first describe a general model algorithm and we state convergence proofs for unspecified scaling parameters satisfying (25.10).

The point z_k^{j+1} , for some j , will be accepted and redefined as x_{k+1} , when a *nonmonotone watchdog test* is satisfied. In this case a watchdog test similar to that considered in Sect. 25.2 can be that of accepting the tentative point z_k^{j+1} if

$$f(z_k^{j+1}) \leq W_k - \max\{\sigma_1(\|F(x_k)\|), \sigma_2(\|z_k^{j+1} - x_k\|)\} \quad (25.11)$$

where

$$W_k = \max_{0 \leq j \leq \min(k, M)} \{f(x_{k-j})\}$$

is the reference value and σ_1, σ_2 are forcing functions.

If all tentative points z_k^j for $j = 1, \dots, N$ have been rejected, we backtrack to x_k and we perform a nonmonotone line search along the search direction

$$d_k = -\frac{1}{\mu_k^0} F(x_k).$$

A conceptual algorithm model is given below. In this scheme we indicate by $\text{NMDFAGLS}(x, d, \rho, \alpha)$ a call to Algorithm 24.4 of Chap. 24, at a point x for computing a stepsize α along the direction d , with tolerance ρ .

Algorithm 25.2 (Nonmonotone Watchdog Algorithm for Nonlinear Equations)

Data: $x_0 \in R^n$, $\rho_0 \in (0, 1)$, $\theta \in (0, 1)$, integers $N \geq 1$, $M \geq 0$.

Set $watchdog=true$, $k = 0$.

While $\|F(x_k)\| \neq 0$ **do**

 1. Set $linesearch=true$

 2. **If** $watchdog=true$ **then**

 Set $z_k^0 = x_k$

For $j = 0, 1, N - 1$

 Determine $\mu_k^j \neq 0$, compute the direction $p_k^j = -\frac{1}{\mu_k^j} F(z_k^j)$

 Set $z_k^{j+1} = z_k^j + p_k^j$

If the $watchdog$ test is satisfied, that is, if:

$$f(z_k^{j+1}) \leq W_k - \max\{\sigma_a(\|F(x_k)\|), \sigma_b(\|z_k^{j+1} - x_k\|)\}$$

then set $x_{k+1} = z_k^{j+1}$, $linesearch=false$ and **exit** from Step 2.

End If

End For

End If

 3. **If** $linesearch=true$ **then**

 Set $d_k = p_k^0$

 compute α_k using algorithm NMDFAGLS($x_k, p_k^0, \rho_k, \alpha_k$)

 Set $x_{k+1} = x_k + \alpha_k d_k$ and $\rho_{k+1} = \theta \rho_k$.

If $\alpha_k \neq 0$ **then**

 set $watchdog=true$, $\rho_k = \rho_0$

Else

 set $watchdog=false$

End If

End if

 4. Set $k = k + 1$.

End While

We can note that, in the general case, the algorithm could repeat Step 3 with $\alpha_k = 0$ for all $k \rightarrow \infty$. In this case, the point x_k remains unchanged and the algorithm would fail. This is not surprising, since the system $F(x) = 0$ could have no solution and hence the convergence conditions should be strong enough to imply, at least, the existence of solutions.

In the next proposition we give a sufficient convergence condition.

Proposition 25.2 (Convergence Conditions for Algorithm 25.2) Suppose that the function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable on \mathbb{R}^n and that the level set \mathcal{L}_0 is compact. Let $\{x_k\}$ be the sequence produced by Algorithm 25.2 and suppose that $F(x_k) \neq 0$ for all k . Then the sequence has limit points in \mathcal{L}_0 and we have

$$\lim_{k \rightarrow \infty} F(x_k)^T J(x_k)^T F(x_k) = 0. \quad (25.12)$$

Moreover, if $F(x) \neq 0$ implies that $F(x)^T J(x)^T F(x) \neq 0$, we have

$$\lim_{k \rightarrow \infty} F(x_k) = 0, \quad (25.13)$$

and hence every limit point of $\{x_k\}$ is a solution of $F(x) = 0$.

Proof When x_{k+1} is computed at step 2 during the watchdog phase, we have

$$f(x_{k+1}) \leq W_k - \sigma_b(\|x_{k+1} - x_k\|), \quad (25.14)$$

while, when x_{k+1} is obtained through the line search, we obtain

$$f(x_{k+1}) \leq W_k - \gamma \|x_{k+1} - x_k\|^2. \quad (25.15)$$

Then the assumptions of Proposition 24.3 are satisfied with an appropriate choice of the forcing function, say for $\sigma(t) \equiv \min\{\sigma_b(t), \gamma t^2\}$, and hence the assertions of this proposition must hold. In particular, we have that $\{x_k\}$ remains in \mathcal{L}_0 and has limit points (by compactness of \mathcal{L}_0), that the sequences $\{W_k\}$ and $\{f(x_k)\}$ converge to the same limit and that $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$.

In order to establish our thesis, let us assume, by contradiction, that there exists an infinite subset of indices K such that $x_k \rightarrow \hat{x}$, $k \in K$ and that

$$\lim_{k \in K, k \rightarrow \infty} F(x_k)^T J(x_k)^T F(x_k) = F(\hat{x})^T J(\hat{x})^T F(\hat{x}) \neq 0. \quad (25.16)$$

Now, suppose first that there exists an infinite subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$, such that x_{k+1} is obtained through the line search NMDFAGLS. It easily verified

that now all the assumptions of Proposition 24.10 are satisfied and hence we must have

$$\begin{aligned} \lim_{k \in K_1, k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} &= \lim_{k \in K_1, k \rightarrow \infty} \frac{F(x_k)^T J(x_k)^T F(x_k)}{\|F(x_k)\|} \\ &= \frac{F(\hat{x})^T J(\hat{x})^T F(\hat{x})}{\|F(\hat{x})\|} = 0, \end{aligned}$$

which contradicts (25.16). Now assume that, for all sufficiently large $k \in K$, the point x_{k+1} is accepted at step 2, because of the fact that the watchdog test is satisfied. This implies, in particular, that

$$f(x_{k+1}) \leq W_k - \sigma_a(\|F(x_k)\|)$$

As $\{W_k\}$ and $\{f(x_k)\}$ converge to the same limit, we have $\lim_{k \in K_1, k \rightarrow \infty} F(x_k) = 0$, which yields again a contradiction to (25.16). Thus (25.12) is true and the last assertion is an obvious consequence. \square

An immediate consequence of the preceding proposition is the following corollary.

Corollary 25.1 (Convergence Conditions for Algorithm 25.2) Suppose that the function $f : R^n \rightarrow R$ is continuously differentiable on R^n , that the level set \mathcal{L}_0 is compact and that the Jacobian matrix is positive (or negative) definite on \mathcal{L}_0 . Let $\{x_k\}$ be the sequence produced by Algorithm 25.2 and suppose that $F(x_k) \neq 0$ for all k .

Then, the sequence has limit points in \mathcal{L}_0 , we have

$$\lim_{k \rightarrow \infty} F(x_k) = 0, \quad (25.17)$$

and every limit point of $\{x_k\}$ is a solution of the system. \square

In the general case, even if the Jacobian matrix is non singular on a compact level set \mathcal{L}_0 , we cannot guarantee that the limit points of the sequence produced by Algorithm 25.2 are stationary points of the merit function f , which obviously is a necessary condition to be solutions of the system. However, it is possible to construct a hybrid algorithm (see, e.g. [135]), employing, when required, linesearches along a set of coordinate directions, which guarantees convergence towards stationary points of f . This implies that the modified algorithm will converge towards solutions of the system when J is non singular on a compact level set.

25.4 Projected Spectral Gradient Method for Minimization on Convex Sets

The spectral gradient method introduced in the unconstrained case has been extended to the minimization of smooth functions on a convex set, and, in particular, to minimization problems with box constraints, by modifying the projection algorithm, which has been described in Chap. 20.

The simplest technique can be one proposed in [21], which consists, in essence, in employing the BB step-size along the steepest descent direction for generating the point to be projected in the convex feasible set S . Then a search direction is computed and a nonmonotone line search is carried out.

More specifically, at each iteration k we compute initially the safeguarded BB scaling factor $1/\mu_k$ in a way that condition (25.8) is satisfied. Letting $s_k = 1/\mu_k$, we determine the point $x_k - s_k \nabla f(x_k)$ and the projection of this point onto S , that is

$$\hat{x}_k = p[x_k - s_k \nabla f(x_k)],$$

where $p[\cdot]$ denotes the projection operator. Then, we can define the feasible direction

$$d_k = \hat{x}_k - x_k$$

and we can perform a nonmonotone line search along d_k . It can be easily shown (see Chap. 20) that, if x_k is not a critical point, the feasible direction d_k is a descent direction satisfying $\nabla f(x_k)^T d_k < 0$.

Thus we can define the following conceptual scheme.

Algorithm 25.3 (Spectral Gradient Projection Method)

Choose the initial point $x_0 \in R^n$, and a scalar $\mu_0 > 0$.

For k=0,1,...

2. Compute the safeguarded BB scaling factor μ_k and set $s_k = 1/\mu_k$
3. Compute the point $x_k - s_k \nabla f(x_k)$ and the projection

$$\hat{x}_k = p[x_k - s_k \nabla f(x_k)].$$

If $\hat{x} = x_k$ **terminate**; otherwise, set $d_k = \hat{x}_k - x_k$.

3. Compute $\alpha_k \in (0, 1] > 0$ by the nonmonotone Armijo line search.
4. Set $x_{k+1} = x_k + \alpha_k d_k$.

End For

We can state the following convergence result, which is proved in [21].

Proposition 25.3 (Convergence of the Spectral Gradient Method) *Let $f : R^n \rightarrow R$ be continuously differentiable over an open set containing the compact convex set S . Let $\{x_k\}$ be the sequence generated by Algorithm 25.3. Then, either there exists an index $v \geq 0$ such that x_v is a critical point, or an infinite sequence is generated and each accumulation point of $\{x_k\}$ is a critical point.*

25.5 Exercises

25.1 Give a convergence proof for Algorithm 25.1.

25.2 Define a computer code for Algorithm 25.1 and perform numerical experiments on standard test problems.

25.3 Define a computer code for Algorithm 25.2 and perform numerical experiments on standard test problems.

25.6 Notes and References

The *spectral* (or *Barzilai-Borwein*)*gradient method* is still the object of a very large number of works, concerning the choice of the step-length μ_k , the study of the convergence rate, the choice of the (typically nonmonotone) linesearch, the application to nonlinear equations and to constrained problems.

In the quadratic case it has been shown that the convergence rate is at least R -linear [55], but it can be Q -linear under suitable limitations on the spectrum of the matrix Q .

Extensions to algorithms employing delayed choices of the step-length have been studied in [101]. Various alternative proposals have been suggested for the computation of the step-size and for the choice of the linesearch. We mention, in particular the papers [261] and [6], concerned with the quadratic case.

The work of Raydan on the application of the BB method to non quadratic problems (the GBB method [224] mentioned before) has revealed the efficiency of the method in large scale problem, if the method is properly implemented, through the use of a non monotone linesearch. This has motivated many different studies and extensions. The algorithm proposed here has been introduced in [134] and has been largely experimented in computational applications. The extension to residual-based

methods for non linear equations has been proposed in [160] (using finite-difference approximations) and in [159] by employing a derivative-free linesearch based on [163]. The technique proposed here is derived from the globalization algorithm given in [135].

The extension to constrained problems has been considered in several papers and it would be impossible, for space limitation, to report here a significant account of the various contributions. We mention the papers [56, 94] and the paper [82] that contains an ample review of the papers on projected spectral gradient methods published up to 2015.

Chapter 26

Decomposition Methods



Decomposition techniques are employed in the solution of optimization problems when, fixing some block of variables, we obtain subproblems of smaller dimension and often of a simpler structure, which can be solved through specialized techniques. Thus, by alternating the minimization with respect to different blocks we can often obtain a more efficient solution technique.

In the sequel we first illustrate some motivations for adopting a decomposition strategy and we report a few significant examples. Then we define our notation, we describe the main classes of decomposition strategies, we give basic convergence results for the best-known techniques and we define some implementable solution algorithms for unconstrained and constrained problems.

26.1 Motivation and Examples

Typical motivations for the use of a decomposition strategy can be the following.

- (a) If the problem dimension is very large, it could be difficult, or even impossible, to solve the problem with standard algorithms, because of limitations of the “working memory” in the computer employed. In this case, we must necessarily define subproblems of smaller dimension, which extract the data from a “mass storage”, when required.
- (b) When a block of (few) variables is fixed, in some cases we can obtain separable problems in the remaining variables, which can be solved through parallel computing techniques.
- (c) By an appropriate choice of the decomposition, we often can obtain many subproblems that can be solved efficiently (locally or globally) and possibly even analytically. In particular, in some non convex problems there is a set of subproblems which are convex with respect to the block of variables associated to each of them and we can find or approximate a global solution. This can

be more advantageous than applying local algorithms to the original non-decomposed problem.

- (d) By introducing auxiliary variables and employing penalization techniques, we can often transform the minimization with incremental methods, which correspond to a *function decomposition* of a composite objective function, into an equivalent block decomposition in terms of variables.

Although decomposition of optimization problems is very useful in many instances, we must devise suitable strategies for guaranteeing global convergence and this could be quite difficult in the non convex case. Moreover, the decomposition can impair the convergence rate, in comparison with that of standard methods.

In practice, we must evaluate the convenience of adopting decomposition methods for each specific class of optimization problem, by estimating, whenever possible, (for instance, through computer experimentation on test problems) the time required for the solution of each subproblem and the iterations required for convergence, in comparison with the computational cost of solving the original problem without decomposition.

Example 26.1 Consider the problem of minimizing an objective function of the form

$$f(x) \equiv \psi_1(x_1) + \sum_{i=2}^m \psi_i(x_1)\phi_i(x_i)$$

over the Cartesian product

$$X = \prod_{i=1}^m X_i,$$

where $x_i \in X_i \subseteq R^{n_i}$, $\sum_{i=1}^m n_i = n$, $\psi_i : X_1 \rightarrow R$, for $i = 1, \dots, m$, $\phi_i : R^{n_i} \rightarrow R$, for $i = 2, \dots, m$ and $f : X \rightarrow R$.

We note that when the block x_1 is fixed, we obtain a function which is *separable* in the remaining blocks, and hence it can be minimized in *parallel*. In this case, under suitable assumptions (in particular, a typical assumption is that the sets X_i are closed convex sets and that the subproblems admit an optimal solution), every major iteration k could be performed as in the following conceptual model.

1. Given the current point x^k , determine a solution x_1^* of the n_1 -dimensional problem

$$\min_{x_1 \in X_1} \psi_1(x_1) + \sum_{i=2}^m \psi_i(x_1)\phi_i(x_i^k),$$

and set $x_1^{k+1} = x_1^*$.

2. For $i = 2, \dots, m$, determine *in parallel* the solutions x_i^* of the n_i -dimensional subproblems

$$\min_{x_i \in X_i} \psi_i(x_1^{k+1})\phi_i(x_i),$$

and set $x_i^{k+1} = x_i^*$.

3. Set $k = k + 1$.

The potential advantages of a decomposition algorithm can be due to the fact that at each iteration we must solve m subproblems of smaller dimension and $m - 1$ of these problems can be solved in parallel. \square

Example 26.2 Consider the nonlinear least squares problem of the form

$$\min_{x,y} f(x, y) \equiv \frac{1}{2} \|\Phi(y)x - b\|^2,$$

where $\Phi \in R^{m \times n}$ is a matrix function, depending on $y \in R^p$, and we have $x \in R^n$, $b \in R^m$. The variables of our problem are then the components of the vector $(x^T y^T)^T \in R^{n+p}$. We note that, if y is fixed, the subproblem in x is a linear least squares problem that can be solved efficiently by employing direct or iterative methods.

In this case, a convenient decomposition scheme could be the *two-block* decomposition at each main iteration, reported below.

1. Given the current point (x^k, y^k) , compute a solution x^* of the *linear* least squares problem

$$\min_x f(x, y^k) \equiv \frac{1}{2} \|\Phi(y^k)x - b\|^2,$$

and set $x^{k+1} = x^*$.

2. Compute a solution y^* of the *non linear* least squares problem

$$\min_y f(x^{k+1}, y) \equiv \frac{1}{2} \|\Phi(y)x^{k+1} - b\|^2,$$

and set $y^{k+1} = y^*$.

3. Set $k = k + 1$. \square

Example 26.3 Suppose we have the problem

$$\min_x f(x) \equiv f(x_1, x_2, \dots, x_n) \equiv \sum_{i=1}^n (x_i - 1)^2 + 4 \prod_{i=1}^n x_i + \prod_{i=1}^n x_i^2,$$

where the non convex objective function $f : R^n \rightarrow R$ is *componentwise strictly convex*, in the sense that when $n - 1$ components of x are fixed, the function in the

remaining variable is strictly convex. In particular, in our example, the solution of each subproblem can be computed analytically. A simple decomposition scheme at each main iteration k can be the following.

1. Given the current point x^k , for $i = 1, \dots, n$ determine analytically the solution x_i^* of the quadratic subproblem

$$\min_{x_i} f(x_1^{k+1}, x_2^{k+1}, \dots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \dots, x_n^k)$$

and set $x_i^{k+1} = x_i^*$.

2. Set $k = k + 1$.

Actually this scheme is essentially a *coordinate search algorithm* with *exact line searches*. In this case, the decomposition can be advantageous not only because the subproblems can be solved analytically, but also from the point of view of global optimization, as the exact *global* minimization along the coordinate axes can help escaping from *local* minimizers. \square

26.2 Classes of Decomposition Methods and Basic Notation

Decomposition methods are typically organized into a sequence of iterations, which generate a sequence of points $\{x^k\}$ that should converge, in the limit, to a critical point of a given optimization problem. We suppose that our problem is of the form

$$\min f(x), \quad x \in X, \tag{26.1}$$

where $f : R^n \rightarrow R$ is continuously differentiable and X is a non empty closed convex set. From the results of Chap. 4 we know that a critical point of problem (26.1) is a point $\bar{x} \in X$, such that

$$\nabla f(\bar{x})^T(y - \bar{x}) \geq 0 \quad \text{for all } y \in X.$$

The properties and the behaviour of a decomposition method depend on some basic choices:

- the kind of decomposition of the vector x into different blocks and the corresponding definition of the subproblems;
- the sequencing of the subproblems during the iterations and the criterion used at each major step k , for choosing the blocks to be updated and for defining the next iterate x^{k+1} ;
- the algorithms employed for solving (exactly or approximately) the subproblems that have been selected.

As regards the first point, we must distinguish between two kinds of decompositions. In the first one, we suppose that x is partitioned into $m \leq n$ distinct blocks, such that each scalar component of x appears in one and only one block, and we have $X = \prod_{i=1}^m X_i$, $x_i \in X_i \subseteq R^{n_i}$, $\sum_{i=1}^m n_i = n$. The second possibility is that we may have overlapping blocks, such that some scalar components of x can appear in two or more different blocks. In this case we have obviously that $\sum_{i=1}^m n_i \geq n$.

Another important distinction, which is related to the sequencing of the subproblems, concerns the dependency of the decomposition on the evolution of the algorithm, in the sense that the decomposition into different blocks can be constant with k , or it can be modified at each major iteration. In the latter case, at each k we must define the block of variables that are updated (the so-called *working set*) and those that remain fixed at their current value. This is obviously a quite general kind of decomposition and we can include in this framework the decomposition with overlapping blocks.

Now, consider first, in more detail, the case of constant prefixed decomposition without overlapping blocks. Under these assumptions, a non ambiguous representation of x is given by $x = (x_1, \dots, x_i, \dots, x_m)$, where $x_i \in R^{n_i}$. When convenient, the objective function value $f(x)$ will be indicated also with $f(x_1, \dots, x_i, \dots, x_m)$. We denote by $\nabla_i f(x) \equiv \nabla_i f(x_1, \dots, x_i, \dots, x_m) \in R^{n_i}$ the partial gradient of f with respect to x_i .

If both \bar{x} and y are partitioned into the same number of block components such that $x_i, y_i \in R^{n_i}$, then it is easily seen that \bar{x} is a critical point of problem (26.1) if and only if we have

$$\nabla_i f(\bar{x})^T (y_i - \bar{x}_i) \geq 0, \quad \text{for all } y_i \in X_i, i = 1, \dots, m.$$

The algorithms with constant decomposition can be further distinguished in two basic schemes, in dependency of the sequencing of the subproblems:

- *sequential* decomposition;
- *parallel* decomposition.

In sequential algorithms, at each major iteration, indexed by k , starting from the point $x^k = (x_1^k, \dots, x_i^k, \dots, x_m^k)$, we perform, in sequence, m inner steps and at each of these steps we update one of the block components. Thus, at the i -th local step we obtain the point

$$(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \dots, x_m^k),$$

where the components with indices $1, \dots, i$ have been updated, while the remaining components are the same as in x^k .

In the parallel decomposition schemes considered here, we suppose that, starting from the point $x^k = (x_1^k, \dots, x_i^k, \dots, x_m^k)$, the m blocks are updated in parallel and independently for $i = 1, \dots, m$, thus determining the tentative components u_i^{k+1} and hence obtaining the points $w(k, i) = (x_1^k, \dots, u_i^{k+1}, \dots, x_m^k)$, $i = 1, \dots, m$.

Then, the next point x^{k+1} is constructed starting from these points, through some rule that may consist, for instance, in choosing as next point x^{k+1} the tentative point that yields the lowest value of the objective function.

Algorithms with variable decomposition and also those with possible overlapping between different blocks, can be more conveniently described by introducing an index set (the working set) that will be denoted by $W^k \subseteq \{1, \dots, n\}$, which identifies the scalar components of x^k that are updated at step k . The complement set, that is the index set of the scalar components that are not updated at step k , will be denoted by \overline{W}^k . Therefore, at each major iteration k , we can represent the current vector x^k with two blocks, letting

$$x^k = (x_{W^k}^k, x_{\overline{W}^k}^k)$$

and hence the updated point will be defined by

$$x^{k+1} = (x_{W^k}^{k+1}, x_{\overline{W}^k}^{k+1}), \quad \text{where } x_{\overline{W}^k}^{k+1} = x_{\overline{W}^k}^k.$$

We note that, in contrast with the preceding schemes, each major step consists in the solution of a single subproblem and we must specify suitable conditions on the construction of the working set in order to guarantee that all components of x are taken into account.

In all the schemes considered, the “solution” of the subproblems may consist in

- the exact or approximate computation of a global minimizer of f with respect to the i -th block;
- the exact or approximate computation of a critical point of f with respect to the i -th block;
- an exact or inexact line search along a suitable feasible search direction d_i^k , with respect to the i -th block;
- a “null step”, which does not modify the i -th block.

The conditions to be imposed, in order to guarantee global convergence towards a critical point, depend on the assumptions on the problem functions and also on the structure of the decomposition into blocks. These conditions will be analyzed in the sequel, with reference to some of the main classes of decomposition schemes. In particular, for problems with constant serial decomposition, we will consider the constrained block version of the *Gauss-Seidel* algorithm, the block *Gauss-Southwell* algorithm and the *block descent* algorithms. In the case of parallel decomposition we will confine ourselves to study some *Jacobi*-type algorithms. Methods with variable decomposition, based on working sets, will be considered in connection with the case of overlapping blocks and with decomposition techniques for the interesting class of problems with a single linear equality constraint and box constraints.

26.3 Block Gauss-Seidel (GS) Method and Extensions

In this section we consider the problem

$$\min f(x), \quad x \in X, \quad (26.2)$$

where $f : R^n \rightarrow R$ is continuously differentiable and X is given by the Cartesian product of m non empty closed convex sets $X_i \subseteq R^{n_i}$, that is

$$X = \prod_{i=1}^m X_i.$$

26.3.1 The Basic Scheme of the GS Method

The GS method for the solution of problem (26.2) can be viewed as an extension of the GS method for the solution of linear equations and constitutes one of the best known sequential decomposition methods.

We describe the method with the following conceptual scheme, under the assumption that the subproblems admit an optimal solution.

Block Gauss-Seidel (GS) Method

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in X$.

For $k = 0, 1, \dots$

For $i = 1, \dots, m$ compute

$$x_i^{k+1} \in \operatorname{Arg} \min_{\xi \in X_i} f(x_1^{k+1}, \dots, \xi, \dots, x_m^k). \quad (26.3)$$

End For

 Set $x^{k+1} = (x_1^{k+1}, \dots, x_m^{k+1})$.

End For

The GS method is well defined only if the subproblems (26.3) admit an optimal solution. A sufficient condition is that, for some point $x^0 \in X$, the non empty level set defined by $\mathcal{L}_X^0 = \{x \in X : f(x) \leq f(x^0)\}$ is compact.

If the GS method is well defined, the global convergence will be established in the sequel in the case of *two-block* decomposition and under some generalized

convexity conditions. In the general, non convex case with $m > 2$ we will consider some modified versions of the GS method and a block-descent method, which does not necessarily require a global minimizations in the component spaces.

We remark that the global convergence of the GS method is not obvious. Indeed, there exist counterexamples showing that the method may fail to converge towards stationary points.

26.3.2 A Line Search Algorithm in the Component Space

In the convergence analysis of the GS method we will make use of a convergence result on an Armijo-type line search in a component space, along a suitable feasible descent direction.

Given a sequence of points $\{v^k\}$ in $X = \prod_{i=1}^m X_i$, let us assume that v^k is partitioned in the form $v^k = (v_1^k, \dots, v_i^k, \dots, v_m^k)$, with $v_i^k \in X_i$.

For a given $i \in (1, \dots, m)$, assume that we can compute a search direction of the form

$$d_i^k = u_i^k - v_i^k, \quad \text{with } u_i^k \in X_i. \quad (26.4)$$

We introduce the following assumption.

Assumption 26.1 Let $\{d_i^k\}$ be a sequence of search directions in X_i defined as in (26.4). Then

- (i) there exists a number $M > 0$ such that $\|d_i^k\| \leq M$ for all k ;
- (ii) we have $\nabla_i f(v^k)^T d_i^k < 0$ for all k . □

We can define the following Armijo-type line search algorithm.

Armijo Line Search (ALS)

Data: $\gamma_i \in (0, 1)$, $\delta_i \in (0, 1)$.

Set $\alpha_i = 1$

While

$$f(v_1^k, \dots, v_i^k + \alpha_i d_i^k, \dots, v_m^k) > f(v^k) + \gamma_i \alpha_i \nabla_i f(v^k)^T d_i^k$$

set $\alpha_i = \delta_i \alpha_i$.

(continued)

End While

Set $\alpha_i^k = \alpha_i$ and **exit**. □

Note that in the preceding algorithm we do not assume that v_i^{k+1} is the result of a line search along d_i^k . Note also, in contrast with usual line search schemes, that the objective function changes at each k , because of the fact that other components could have been updated. However, under Assumption 26.1, we can easily establish that the next proposition holds.

Proposition 26.1 (Convergence of Algorithm ALS) *Let $f : R^n \rightarrow R$ be a continuously differentiable function on an open set containing the closed convex set $X \subseteq R^n$. Suppose that $\{v^k\}$ is a sequence of points in X and let $\{d_i^k\}$ be a sequence of search directions in X_i such that Assumption 26.1 is satisfied. Suppose that α_i^k is computed by means of Algorithm ALS.*

Then:

(a) *the algorithm terminates with a step-size $\alpha_i^k > 0$ satisfying*

$$f(v_1^k, \dots, v_i^k + \alpha_i^k d_i^k, \dots, v_m^k) \leq f(v^k) + \gamma_i \alpha_i^k \nabla_i f(v^k)^T d_i^k;$$

(b) *if $\{v^k\}$ converges to $\bar{v} \in X$ and*

$$\lim_{k \rightarrow \infty} f(v^k) - f(v_1^k, \dots, v_i^k + \alpha_i^k d_i^k, \dots, v_m^k) = 0,$$

we have

$$\lim_{k \rightarrow \infty} \nabla_i f(v^k)^T d_i^k = 0.$$

Proof By assumption, at each k the search direction d_i^k is a feasible descent direction on X_i at the point $\xi = v_i^k$, for the continuously differentiable function of ξ defined by $f(v_1^k, \dots, \xi, \dots, v_m^k)$. Therefore, using standard arguments, it can be easily verified that the Armijo line search determines in a finite number of inner iterations a step-size $\alpha_k \in (0, 1]$ such that (a) holds. We will prove that also (b) is satisfied.

By Assumption 26.1 we have that there exists $M > 0$ such that $0 < \|d_i^k\| \leq M$ for every k ; moreover, since α_i^k is such that condition (a) holds, we can write

$$f(v^k) - f(v_1^k, \dots, v_i^k + \alpha_i^k d_i^k, \dots, v_m^k) \geq \gamma_i \alpha_i^k |\nabla_i f(v^k)^T d_i^k|. \quad (26.5)$$

Therefore, if we assume that v^k converges to $\bar{v} \in X$ and that

$$\lim_{k \rightarrow \infty} f(v^k) - f(v_1^k, \dots, v_i^k + \alpha_i^k d_i^k, \dots, v_m^k) = 0,$$

we obviously obtain

$$\lim_{k \rightarrow \infty} \alpha_i^k |\nabla f(v^k)^T d_i^k| = 0. \quad (26.6)$$

Now, reasoning by contradiction, let us assume that (b) is not true. As $\|d_i^k\|$ is bounded, $\nabla_i f(v^k)^T d_i^k < 0$ and v^k converges to \bar{v} , there must exist a subsequence (relabel it again $\{v^k\}$), a vector $\hat{d}_i \in X_i$ and a positive number η such that

$$\lim_{k \rightarrow \infty} \nabla_i f(v^k)^T d_i^k = \nabla_i f(\bar{v})^T \hat{d}_i = -\eta < 0. \quad (26.7)$$

From (26.6) it follows that

$$\lim_{k \rightarrow \infty} \alpha_i^k = 0 \quad (26.8)$$

and hence, for k sufficiently large, say $k \geq \hat{k}$, we have $\alpha_i^k < 1$. Therefore, for $k \geq \hat{k}$, as the initial step-size has not been accepted, we must necessarily have

$$f(v_1^k, \dots, v_i^k + \frac{\alpha_i^k}{\delta_i} d_i^k, \dots, v_m^k) - f(v^k) > \gamma_i \frac{\alpha_i^k}{\delta_i} \nabla_i f(v^k)^T d_i^k. \quad (26.9)$$

Using the Mean Value Theorem, we can write

$$f(v_1^k, \dots, v_i^k + \frac{\alpha_i^k}{\delta_i} d_i^k, \dots, v_m^k) = f(v^k) + \frac{\alpha_i^k}{\delta_i} \nabla_i f(w(i)^k)^T d_i^k. \quad (26.10)$$

where

$$w(i)^k = (v_1^k, \dots, v_i^k + \theta_i^k \frac{\alpha_i^k}{\delta_i} d_i^k, \dots, v_m^k),$$

with $\theta_i^k \in (0, 1)$, so that $w(i)^k \in X$. Therefore, by (26.9) and (26.10), for $k \geq \hat{k}$, we obtain

$$\nabla_i f(w(i)^k)^T d_i^k > \gamma_i \nabla_i f(v^k)^T d_i^k. \quad (26.11)$$

Since $\|d_i^k\| \leq M$, from (26.8) it follows $\lim_{k \rightarrow \infty} \alpha_i^k \|d_i^k\| = 0$ and hence

$$\lim_{k \rightarrow \infty} w(i)^k = \lim_{k \rightarrow \infty} (v_1^k, \dots, v_i^k + \theta_i^k \frac{\alpha_i^k}{\delta_i} d_i^k, \dots, v_m^k) = \bar{v}.$$

As a consequence, taking limits for $k \rightarrow \infty$, we get $\nabla_i f(\bar{v})^T \hat{d} \geq \gamma \nabla_i f(\bar{v})^T \hat{d}$. This condition, taking into account (26.7), implies $\eta \leq \gamma \eta$, which contradicts the assumption $\gamma < 1$. Then we can conclude that (26.7) leads to a contradiction, so that (b) must hold. \square

26.3.3 Limit Points of the GS Algorithm

Consider now the GS algorithm. We give here some preliminary results on the limit points of the sequences generated by the GS algorithm.

In order to simplify notation, we indicate by $z(k, i)$ a vector such that $z(k, 0) \equiv x^k$ and, for $i = 1, \dots, m$:

$$z(k, i) = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \dots, x_m^k).$$

Thus, at the end of the m -th step, we have $x^{k+1} = z(k, m) = z(k + 1, 0)$.

For convenience, we set also $z(k, m+1) = z(k+1, 1)$. As $z(k, i)$ is a constrained global minimizer of $f(x_1^{k+1}, \dots, x_{i-1}^{k+1}, \xi, x_{i+1}^k, \dots, x_m^k)$, with respect to $\xi \in X_i$, from the results of Chap. 4 we have that, for each $i \in \{1, \dots, m\}$, we have

$$\nabla_i f(z(k, i))^T (y_i - x_i^{k+1}) \geq 0 \quad \text{for all } y_i \in X_i. \quad (26.12)$$

Some basic properties of the sequences generated by the GS method are stated in the next proposition.

Proposition 26.2 *Let $f : R^n \rightarrow R$ be a continuously differentiable function on an open set containing the closed convex set $X \subseteq R^n$ and assume that the level set \mathcal{L}_X^0 is compact. For $i = 1, \dots, m$ let $\{z(k, i)\}$ be the sequences generated by the GS method. Then:*

- (i) *we have $z(k, i) \in \mathcal{L}_X^0$ for every k and every $i = 1, \dots, m$;*
- (ii) *the sequences $\{f(z(k, i))\}$ for $i = 1, \dots, m$ converge to the same limit;*
- (iii) *for every $i = 1, \dots, m$ the sequence $\{z(k, i)\}$ has limit points and, if \hat{z} is a limit point of $\{z(k, i)\}$, we have*

$$\nabla_i f(\hat{z})^T (y_i - \hat{z}_i) \geq 0 \quad \text{for all } y_i \in X_i, \quad (26.13)$$

and also

$$\nabla_{i^*} f(\hat{z})^T (y_{i^*} - \hat{z}_{i^*}) \geq 0 \quad \text{for all } y_{i^*} \in X_{i^*}, \quad (26.14)$$

where $i^ = i + 1$ if $i \leq m - 1$ and $i^* = 1$ if $i = m$.*

Proof The instructions of the algorithm imply that all points $z(k, i)$, $i = 1, \dots, m$ are feasible and that

$$f(x^0) \geq f(x^k) \geq f(z(k, 1)) \geq \dots f(z(k, m)) = f(x^{k+1}).$$

This implies that all the points $z(k, i)$ remain in \mathcal{L}_X^0 and assertion (i) must be true. Then the compactness of the level set implies that f is bounded below on X and therefore the monotone non increasing sequence of function values converges to a unique limit, so that assertion (ii) holds.

In order to prove (iii), we first observe that, by (i) and the compactness of the level set, every sequence $\{z(k, i)\}$ has limit points in X . If \hat{z} is a limit point of $\{z(k, i)\}$ then there exists a subsequence $\{z(k, i)_K\}$ converging to \hat{z} . Recalling (26.12), by the continuity of $\nabla_i f$, we get immediately (26.13).

Now, reasoning by contradiction, suppose that (26.14) of assertion (iii) is false. Suppose first that $i \leq m - 1$, so that $i^* = i + 1$ and suppose that there exists $\tilde{y}_{i+1} \in X_{i+1}$ such that

$$\nabla f_{i+1}(\hat{z})^T (\tilde{y}_{i+1} - \hat{z}_{i+1}) < 0. \quad (26.15)$$

Now, for $k \in K$, let

$$d_{i+1}^k = \tilde{y}_{i+1} - z(k, i)_{i+1} = \tilde{y}_{i+1} - x_{i+1}^k.$$

As $\{z(k, i)\}_K$ converges to \hat{z} , it follows that the sequence $\{d_{i+1}^k\}_K$ is bounded. Recalling (26.15), it follows that there exists an infinite subset $K_1 \subseteq K$, such that

$$\nabla f_{i+1}(z(k, i))^T d_{i+1}^k < 0, \quad \text{for all } k \in K_1. \quad (26.16)$$

Thus the sequences $\{z(k, i)\}_{K_1}$ and $\{d_{i+1}^k\}_{K_1}$ are such that Assumption 26.1 is satisfied provided that we identify $\{v^k\}$ with $\{z(k, i)\}_{K_1}$.

Now, for $k \in K_1$ suppose that we compute α_{i+1}^k by employing the line search algorithm ALS. Then we have

$$f(x_1^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k + \alpha_{i+1}^k d_{i+1}^k, \dots, x_m^k) \leq f(z(k, i)).$$

Since $x_{i+1}^k \in X_{i+1}$, $x_{i+1}^k + d_{i+1}^k \in X_{i+1}$ and $\alpha_{i+1}^k \in (0, 1]$, by convexity of X_{i+1} we have

$$x_{i+1}^k + \alpha_{i+1}^k d_{i+1}^k \in X_{i+1}.$$

As

$$f(z(k, i+1)) = \min_{\xi \in X_{i+1}} f(x_1^{k+1}, \dots, x_i^{k+1}, \xi, \dots, x_m^k),$$

we can write

$$\begin{aligned} f(z(k, i+1)) &\leq f(x_1^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k + \alpha_{i+1}^k d_{i+1}^k, \dots, x_m^k) \\ &\leq f(z(k, i)). \end{aligned} \tag{26.17}$$

By (ii) the sequences $\{f(z(k, j))\}$ converge the same limit for all $j \in \{1, \dots, m\}$ and hence we must have, in particular:

$$\lim_{k \in K_1, k \rightarrow \infty} f(z(k, i)) - f(x_1^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k + \alpha_{i+1}^k d_{i+1}^k, \dots, x_m^k) = 0.$$

Then, by Proposition 26.1, where we identify $\{v^k\}$ with $\{z(k, i)\}_{K_1}$, we have

$$\nabla f_{i+1}(\hat{z})^T (\tilde{y}_{i+1} - \hat{z}_{i+1}) = 0,$$

which contradicts (26.15), so that (ii) holds for $i = 1, \dots, m-1$. If $i = m$ and $i^* = 1$, we can repeat the same reasoning, noting that $z(k, m+1) = z(k+1, 1)$. \square

In the general case, the preceding proposition does not guarantee the convergence of the GS method to critical points of f , as the sequences $\{z(k, i)\}$ may have different limit points for different values of i . In particular, even if (26.14) is satisfied, the point \hat{z} is not necessarily a limit point of $\{z(k, i^*)\}$. However, in the case of $m = 2$ (*two-block GS method*), convergence follows immediately from the preceding result.

26.3.4 The Two-Block GS Method

Consider the problem

$$\begin{aligned} \min_{x_1, x_2} f(x_1, x_2), \\ x_1 \in X_1, x_2 \in X_2 \end{aligned}$$

where $X_1 \subseteq R^{n_2}$, $X_2 \subseteq R^{n_2}$. An interesting special case has been given in Example 26.2. We can state the following result.

Proposition 26.3 (Convergence of the Two-Block GS Method) Suppose that the assumptions of Proposition 26.2 are satisfied and that we have $m = 2$. Then, the sequence $\{x^k\}$ generated by the GS method has limit points and every limit point is a critical point of the problem.

Proof We have $z(k - 1, 2) = x^k$ and hence, by (iii) of Proposition 26.2 the sequence $\{x^k\}$ has limit points. If \bar{x} is a limit point there exists a subsequence $\{x^K\}_K$ converging to \bar{x} and hence $\{z(k - 1, 2)\}_K$ converges to \bar{x} . By Proposition 26.2, this implies that

$$\nabla f_2(\bar{x})^T(y_2 - \bar{x}_2) \geq 0 \quad \text{for all } y_2 \in X_2$$

and also, as $i^* = 1$, that

$$\nabla f_1(\bar{x})^T(y_1 - \bar{x}_1) \geq 0 \quad \text{for all } y_1 \in X_1$$

Thus, \bar{x} is a critical point of the problem. \square

26.3.5 Convergence of the GS Method Under Generalized Convexity Assumptions

When $m > 2$ convergence of the GS method can be established under some generalized convexity assumptions or uniqueness conditions on the subproblems. First we consider the case of a pseudoconvex objective function. We recall from Appendix C the definition of pseudoconvex function.

Definition 26.1 (Pseudoconvex Function) Let $D \subseteq R^n$ be an open set, let S be a convex set contained in D , let $f : D \rightarrow R$ and assume that ∇f is continuous on D .

Then f is said to be *pseudoconvex* on S if, for all pairs $x, y \in S$, we have that $\nabla f(x)^T(y - x) \geq 0$ implies $f(y) \geq f(x)$. \square

We state the following proposition.

Proposition 26.4 (Convergence of the GS Method for Pseudoconvex f)
Suppose that the assumptions of Proposition 26.2 are satisfied and that f is pseudoconvex on X . Then, the sequence $\{x^k\}$ generated by the GS method has limit points and every limit point of $\{x^k\}$ is a global minimizer of f on X .

Proof Consider the updates $z(k, i)$ generated by the GS method, defined in Sect. 26.3.3 for $i = 0, 1, \dots, m$. By Proposition 26.2 the points $z(k, i)$ remain in

the level set \mathcal{L}_X^0 for all i and k , and every subsequence of these points has limit points in \mathcal{L}_X^0 .

Then, if we assume that \bar{x} is a limit point of $\{x^k\}$, we can find an index set K , such that $\{x^k\}_K$ converges to \bar{x} and that, for the same index set, the subsequences $\{z(k, i)\}_K$, for $i = 1, \dots, m$, converge to limit points $\bar{z}^{(i)} \in \mathcal{L}_X^0$.

Now, for $i = 1, \dots, m$, we can write:

$$z(k, i) = z(k, i - 1) + d(k, i),$$

where $d(k, i) \in R^n$ is such that the components $d_h(k, i) \in R^{n_h}$ with $h \neq i$ are zero. Therefore, taking limits for $k \in K$, we get, for $i = 1, \dots, m$

$$\bar{z}^{(i)} = \bar{z}^{(i-1)} + \bar{d}^{(i)}, \quad (26.18)$$

where

$$\bar{d}^{(i)} = \lim_{k \in K, k \rightarrow \infty} d(k, i), \quad d_h^{(i)} = 0, \quad h \neq i.$$

By Proposition 26.2 we have

$$f(\bar{x}) = f(\bar{z}^{(i)}), \quad \text{for } i = 1, \dots, m \quad (26.19)$$

and also:

$$\nabla_i f(\bar{z}^{(i)})^T (y_i - \bar{z}_i^{(i)}) \geq 0, \quad \text{for all } y_i \in X_i, \quad (26.20)$$

$$\nabla_{i^*} f(\bar{z}^{(i)})^T (y_{i^*} - \bar{z}_{i^*}^{(i)}) \geq 0, \quad \text{for all } y_{i^*} \in X_{i^*}, \quad (26.21)$$

where $i^* = i + 1$ if $i \leq m - 1$ and $i^* = 1$ if $i = m$.

Now, we show first that, given $j, \ell \in \{1, \dots, m\}$ such that

$$\nabla_\ell f(\bar{z}^{(j)})^T (y_\ell - \bar{z}_\ell^{(j)}) \geq 0, \quad \text{for all } y_\ell \in X_\ell, \quad (26.22)$$

then we have also

$$\nabla_\ell f(\bar{z}^{(j-1)})^T (y_\ell - \bar{z}_\ell^{(j-1)}) \geq 0, \quad \text{for all } y_\ell \in X_\ell. \quad (26.23)$$

If $\ell = j$ then (26.23) holds because of (26.21), where we set $i^* = j$ and $i = j - 1$. Then we can assume $\ell \neq j$. By (26.18) we have

$$\bar{z}^{(j)} = \bar{z}^{(j-1)} + \bar{d}^{(j)}, \quad (26.24)$$

where $d_h^{(j)} = 0$ for $h \neq j$.

For any given vector $\eta \in R^{n_\ell}$, such that $\bar{z}_\ell^{(j-1)} + \eta \in X_\ell$, we can define the feasible point

$$u(\eta) = \bar{z}^{(j-1)} + d(\eta),$$

where $d_h(\eta) = 0$ for $h \neq \ell$ and $d_\ell(\eta) = \eta$.

Then, from (26.20) and (26.22), noting that (26.24) and the assumption $\ell \neq j$ imply

$$\eta = u_\ell(\eta) - \bar{z}_\ell^{(j-1)} = u_\ell(\eta) - \bar{z}_\ell^{(j)},$$

we obtain

$$\begin{aligned} \nabla f(\bar{z}^{(j)})^T(u(\eta) - \bar{z}^{(j)}) &= \nabla f(\bar{z}^{(j)})^T(\bar{z}^{(j-1)} + d(\eta) - \bar{z}^{(j)}) \\ &= \nabla_j f(\bar{z}^{(j)})^T(\bar{z}^{(j-1)} - \bar{z}^{(j)}) + \nabla_\ell f(\bar{z}^{(j)})^T \eta \\ &= \nabla_j f(\bar{z}^{(j)})^T(\bar{z}^{(j-1)} - \bar{z}^{(j)}) + \nabla_\ell f(\bar{z}^{(j)})^T \\ &= (u_\ell(\eta) - \bar{z}_\ell^{(j)}) \geq 0. \end{aligned}$$

It follows, by the pseudoconvexity of f , that

$$f(u(\eta)) \geq f(\bar{z}^{(j)}) \quad \text{for all } \eta \in R^{n_\ell} \text{ such that } \bar{z}_\ell^{(j-1)} + \eta \in X_\ell$$

On the other hand, as $f(\bar{z}^{(j)}) = f(\bar{z}^{(j-1)})$, we obtain

$$f(u(\eta)) \geq f(\bar{z}^{(j-1)}) \quad \text{for all } \eta \in R^{n_\ell} \text{ such that } \bar{z}_\ell^{(j-1)} + \eta \in X_\ell,$$

which by definition of $u(\eta)$, implies (26.23).

Finally, as we have proved that (26.22) implies (26.23), by induction we obtain, for every $j \in \{1, \dots, m\}$

$$\nabla_j f(\bar{z}^{(0)})^T(\bar{y}_j - \bar{z}_j^{(0)}) = \nabla_j f(\bar{x})^T(y_j - \bar{x}_j) \geq 0 \quad \text{for all } y_j \in X_j,$$

which proves the thesis. \square

When f is not pseudoconvex and $m > 2$, we can establish the convergence of the GS method by imposing stronger requirements on the subproblems or by introducing suitable modifications in the basic scheme.

A well known result (for f continuously differentiable on X) is that, if *the solution of the subproblems is unique*, for every $i = 1, \dots, m$ then every limit point of the sequence $\{x^k\}$ generated by the GS method is a critical point. A proof can be found, for instance, in [16].

Here we report a result which consists in imposing a *strong quasi-convexity* assumption on the objective function, with respect to $m - 2$ block components.

First we recall, as follows, the definition of strong quasi-convexity, with reference to a given $i \in \{1, \dots, m\}$.

Definition 26.2 We say that f is strongly quasi-convex on X , with respect to $x_i \in X_i$, if, for every $x \in X$ and $y_i \in X_i$, with $y_i \neq x_i$, we have

$$\begin{aligned} & f(x_1, \dots, (tx_i + (1-t)y_i), \dots, x_m) \\ & < \max\{f(x), f(x_1, \dots, y_i, \dots, x_m)\} \text{ for all } t \in (0, 1) \end{aligned}$$

□

Preliminarily, we need the following result, where we assume that $X = \prod_{i=1}^m X_i$ and that all vectors in X are decomposed into block-components.

Proposition 26.5 Let $f : X \rightarrow R$ be strongly quasi-convex on X with respect to $x_i \in X_i$ in the sense of Definition 26.2. Let $\{y^k\}$ be a sequence on X , with block-components $y_j^k \in X_j$ for $j = 1, \dots, m$, converging to $\bar{y} \in X$ and let $\{v^k\}$ a sequence of vectors in X with block-components $v_j^k \in X_j$ for $j = 1, \dots, m$, defined by:

$$v_i^k \in \operatorname{Argmin}_{\xi \in X_i} f(y_1^k, \dots, y_{i-1}^k, \xi, \dots, y_m^k)$$

$$v_j^k = y_j^k, \quad j \neq i.$$

Then, if $\lim_{k \rightarrow \infty} f(y^k) - f(v^k) = 0$ we have

$$\lim_{k \rightarrow \infty} \|v_i^k - y_i^k\| = 0.$$

Proof Reasoning by contradiction, suppose there exists a subsequence $\{y^k\}_K$ and a number $\beta > 0$ such that

$$\|v^k - y^k\| = \|v_i^k - y_i^k\| \geq \beta, \quad k \in K. \quad (26.25)$$

For each $k \in K$, let $s^k = (v^k - y^k)/\|v^k - y^k\|$ and take $\lambda \in (0, 1)$. Then, the point

$$\tilde{v}^k = y^k + \lambda \beta s^k,$$

is in the line segment $[y^k, v^k] \subseteq X$. As $\{y^k\}$ converges to $\bar{y} = (\bar{y}_1, \dots, \bar{y}_i, \dots, \bar{y}_m)$, and $\|s^k\| = 1$, there must exist a subsequence with $K_1 \subseteq K$ such that $\{\tilde{v}^k\}_{K_1}$ converges to a point $\bar{y}^* = (\bar{y}_1, \dots, \bar{y}_i^*, \dots, \bar{y}_m) \in X$, satisfying $\|\bar{y}_i - \bar{y}_i^*\| = \lambda\beta > 0$.

By the assumption of strong quasi-convexity with respect to the i -th component and the definition of v^k , we have, for all $t \in (0, 1)$:

$$\begin{aligned} f(y^k) &= \max\{f(y^k), f(v^k)\} > f(y_1^k, \dots, (1-t)y_i^k + t(y_i^k + \lambda\beta s_i^k), \dots, y_m^k) \\ &\geq f(v^k) \end{aligned}$$

As, by assumption, we have $\lim_{k \rightarrow \infty} f(y^k) - f(v^k) = 0$, taking limits for $k \in K_1$, we obtain:

$$f(\bar{y}) = f(\bar{y}_1, \dots, (1-t)\bar{y}_i + t\bar{y}_i^*, \dots, \bar{y}_m), \quad \text{for all } t \in (0, 1),$$

which contradicts the strong quasi-convexity assumption with respect to y_i on X . \square

Then we can state the announced convergence result.

Proposition 26.6 (Convergence of GS Method Under Componentwise Strong Quasi-Convexity Assumption for $m - 2$ Components) Suppose that the assumptions of Proposition 26.2 are satisfied and that, for all $i = 1, \dots, m-2$, the objective function f is componentwise strongly quasi-convex on the closed convex set $X \subseteq R^n$, in the sense of Definition 26.2. Let $\{x^k\}$ be the sequence generated by the GS method. Then, $\{x^k\}$ has limit points and every limit point is a critical point of f on X .

Proof By Proposition 26.2 the sequences generated by the GS method have limit points in \mathcal{L}_X^0 . Now suppose that $\{x^k\}_K$ is a subsequence converging to a limit point $\bar{x} \in R^n$. By Proposition 26.2, we have, in particular, that

$$\lim_{k \rightarrow \infty} f(z(k, i)) - f(\bar{x}) = 0, \quad i = 1, \dots, m.$$

and also that

$$\nabla_m f(\bar{x})^T (y_m - \bar{x}_m) \geq 0, \quad \text{for all } y_m \in X_m. \quad (26.26)$$

By the strong quasi-convexity assumption, from Proposition 26.5, where we identify $\{y^k\}$ with $\{x^k\}_K$ and $\{v^k\}$ with $\{z(k, 1)\}_K$, we obtain $\lim_{k \in K, k \rightarrow \infty} z(k, 1) = \bar{x}$. By repeated application of Proposition 26.5 to the sequences $\{z(k, i-1)\}_K$ and

$\{z(k, i)\}_K$, for $i = 1, \dots, m - 2$, we obtain

$$\lim_{k \in K, k \rightarrow \infty} z(k, i) = \bar{x}, \quad i = 1, \dots, m - 2.$$

Then Proposition 26.2 implies

$$\nabla_i f(\bar{x})^T (y_i - \bar{x}_i) \geq 0, \quad \text{for all } y_i \in X_\ell, \quad i = 1, \dots, m - 1 \quad (26.27)$$

Therefore, the assertion follows from (26.26) and (26.27). \square

26.3.6 Proximal-Point Modification of the GS Method

We consider here a modification of the GS method that can ensure convergence towards critical points of the problem, even if $m > 2$ and generalized convexity assumptions on the objective function or uniqueness conditions on the subproblems are not imposed.

The essence of this modification is that of introducing a *proximal point term* in the solution of the subproblems, in order to penalize large differences among consecutive estimates of the same block of variables.

The scheme defined below is an extension of proximal point modifications proposed in [8] under convexity assumption on f . A conceptual scheme based on this approach is reported below.

Proximal Point Modification of the GS Method (PGS)

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in X$, $\tau_i > 0$, $i = 1, \dots, m$.

For $k = 0, 1, \dots$

For $i = 1, \dots, m$ compute

$$x_i^{k+1} \in \operatorname{Argmin}_{\xi \in X_i} \left\{ f(x_1^{k+1}, \dots, \xi, \dots, x_m^k) + \frac{1}{2} \tau_i \|\xi - x_i^k\|^2 \right\}.$$

End For

Set $x^{k+1} = (x_1^{k+1}, \dots, x_m^{k+1})$.

End For

The convergence of this algorithm is established in the next proposition.

Proposition 26.7 (Convergence of Proximal Point GS Method (PGS))

Suppose that the assumptions of Proposition 26.2 are satisfied. Let $\{x^k\}$ be the sequence generated by the PGS method. Then, $\{x^k\}$ has limit points and every limit point is a critical point of f in X .

Proof By Proposition 26.2 the sequences generated by the PGS method have limit points in \mathcal{L}_X^0 . Let $\{x^k\}_K$ be a subsequence converging to a limit point $\bar{x} \in R^n$. Define the vectors $\tilde{z}(k, 0) = x^k$ and

$$\tilde{z}(k, i) = (x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^{k+1}, x_{i+1}^k, \dots, x_m^k),$$

for $i = 1, \dots, m$. Then, taking $\xi = x_i^k$, we can write

$$f(\tilde{z}(k, i)) \leq f(\tilde{z}(k, i-1)) - \frac{1}{2}\tau_i \|\tilde{z}(k, i) - \tilde{z}(k, i-1)\|^2, \quad (26.28)$$

which implies, for $i = 1, \dots, m$:

$$f(x^{k+1}) \leq f(\tilde{z}(k, i)) \leq f(\tilde{z}(k, i-1)) \leq f(x^k).$$

Reasoning as in Proposition 26.2, we obtain

$$\lim_{k \rightarrow \infty} f(x^{k+1}) - f(x^k) = 0,$$

and hence, taking limits in (26.28) for $k \rightarrow \infty$ we have

$$\lim_{k \rightarrow \infty} \|\tilde{z}(k, i) - \tilde{z}(k, i-1)\|^2 = 0, \quad (26.29)$$

which implies

$$\lim_{k \in K, k \rightarrow \infty} \tilde{z}(k, i) = \bar{x}, \quad i = 0, 1, \dots, m. \quad (26.30)$$

Now, for every $j = 1, \dots, m$ the point $\tilde{z}(k, j)$ must satisfy the optimality condition

$$[\nabla_j f(\tilde{z}(k, j)) + \tau_j (\tilde{z}(k, j) - \tilde{z}(k, j-1))]^T (y_j - \tilde{z}(k, j)) \geq 0, \quad \text{for all } y_j \in X.$$

Then, taking limits for $k \in K, k \rightarrow \infty$, recalling (26.28) and (26.29) and (26.30) we obtain, for every $j = 1, \dots, m$

$$\nabla_j f(\bar{x})^T (y_j - \bar{x}_j) \geq 0 \quad \text{for all } y_j \in X_j,$$

which proves our assertion. \square

26.4 Block Descent Methods

The GS algorithms considered in the preceding section are based on the global solution of the subproblems. This can be too expensive from a computational point of view and, as we know, it does not guarantee convergence in the general case, unless suitable modifications of the GS method are introduced. Thus, if we search for critical points of the original problem, we are interested in block descent methods that can ensure global convergence and make use of standard techniques for *local* minimization. In this section we introduce basic algorithms for unconstrained and constrained optimization.

26.4.1 A Basic Unconstrained Block Descent Algorithm

We consider the unconstrained problem

$$\min f(x) \equiv f(x_1, \dots, x_m), x \in R^n,$$

where x is partitioned into m block components $x_i \in R^{n_i}$, for $i = 1, \dots, m$. We suppose that f is continuously differentiable on R^n and that there exists $x_0 \in R^n$ such that the level set $\mathcal{L}^0 = \{x \in R^n : f(x) \leq f(x_0)\}$ is compact. Under these assumptions, we define the following conceptual model algorithm.

Block Descent (BD) Method

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in R^n$, forcing functions: $\sigma_i : R^+ \rightarrow R^+, i = 1, \dots, m$, scale factors $a_i > 0, i = 1, \dots, m$.

For $k = 0, 1, \dots$

Set $z(k, 0) = x^k$. If $\nabla f(x^k) = 0$ **terminate**.

For $i = 1, \dots, m$

1. Set $d_i^k = -a_i \nabla_i f(z(k, i - 1))$.

2. Set $\alpha_i^k = 0$ if $\nabla_i f(z(k, i - 1)) = 0$, otherwise compute α_i^k along d_i^k using Armijo's method ALS, starting from the tentative step-size $\alpha_i = 1$.

3. Choose x_i^{k+1} in a way that the following conditions are satisfied

$$(3a) \quad f(x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k) \leq f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k);$$

(continued)

(3b) $f(x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k) \leq f(z(k, i-1)) - \sigma_i(\|x_i^{k+1} - x_i^k\|).$

4. Set $z(k, i) = (x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k).$

End For

Set $x^{k+1} = (x_1^{k+1}, \dots, x_m^{k+1}).$

End For

We note that in the preceding scheme it is not specified the criterion used for determining the point x_i^{k+1} . In practice, we may think of some iterations of an efficient unconstrained minimization technique, provided that the conditions of step 3 are satisfied. It can be easily verified that, under the assumption stated, conditions (3a) and (3b) can be satisfied, for instance, if we take x_i^{k+1} as the point obtained through Armijo's line search ALS, along the direction

$$d_i^k = -a_i \nabla_i f(z(k, i-1)),$$

that is

$$x_i^{k+1} = x_i^k + \alpha_i^k d_i^k = x_i^k - \alpha_i^k a_i \nabla_i f(z(k, i-1)).$$

In fact, as $\alpha_i^k \leq 1$, the acceptance criterion in Armijo's method in the component space can be expressed by

$$\begin{aligned} f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) &\leq f(z(k, i-1)) - \gamma_i a_i \alpha_i^k \|\nabla_i f(z(k, i-1))\|^2 \\ &\leq f(z(k, i-1)) - \sigma_i(\|x_i^{k+1} - x_i^k\|), \end{aligned} \tag{26.31}$$

where $\gamma_i \in (0, 1)$ and the forcing function σ_i is defined by $\sigma_i(t) = (\gamma_i/a_i)t^2$, so that conditions (3a) and (3b) are satisfied. Then, we can state the following convergence result.

Proposition 26.8 (Convergence of Block Descent Method (BD)) Suppose that f is continuously differentiable on R^n and that there exists $x^0 \in R^n$ such that the level set $\mathcal{L}^0 = \{x \in R^n : f(x) \leq f(x^0)\}$ is compact. Then, the sequence $\{x^k\}$ generated by Algorithm BD has limit points and every limit point is a critical point of f .

Proof The line search acceptance rule (26.31) implies that we can write, for $i = 1, \dots, m$:

$$f(z(k, i)) \leq f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) \leq f(z(k, i-1)), \quad (26.32)$$

which implies, as $z(k, 0) = x^k$ and $x^{k+1} = z(k, m)$, for $i = 1, \dots, m$,

$$f(x^{k+1}) \leq f(z(k, i)) \leq f(x^k). \quad (26.33)$$

As \mathcal{L}_0 is compact the sequence $\{f(x^k)\}$ is monotone non increasing and bounded below, so that it converges to a limit \tilde{f} . By (26.33) all the sequences $\{f(z(k, i))\}$ for $i = 0, \dots, m$ converge to the same limit, that is

$$\lim_{k \rightarrow \infty} f(z(k, i)) = \tilde{f}, \quad i = 0, \dots, m. \quad (26.34)$$

We have also that all the infinite subsequences of points produced by the method remain in \mathcal{L}_0 and have limit points in \mathcal{L}_0 . As the gradient ∇f is continuous and \mathcal{L}_0 is compact we have that the search directions $d_i^k = -a_i \nabla_i f(z(k, i-1))$, which are obviously feasible in $X_i \equiv R^{n_i}$, are also uniformly bounded.

Now, for some $i \in \{1, \dots, m\}$, suppose that the point \bar{z} is a limit point of $\{z(k, i-1)\}$. From (26.34) and (26.32) it follows that

$$\lim_{k \rightarrow \infty} f(z(k, i-1)) - f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) = 0.$$

Therefore, by Proposition 26.1, by identifying the sequence $\{v^k\}$ considered there with the subsequence of $\{z(k, i-1)\}$ converging to \bar{z} , we obtain

$$\nabla_i f(\bar{z}) = 0, \quad i = 1, \dots, m. \quad (26.35)$$

Then (26.34) and condition (3b) imply

$$\lim_{k \rightarrow \infty} \|z(k, i) - z(k, i-1)\| = 0, \quad i = 1, \dots, m. \quad (26.36)$$

As $x^k = z(k, 0)$, if \bar{x} is a limit point of $\{x^k\}$, Eq. (26.36) implies, by induction, that \bar{x} is also a limit point of $\{z(k, i)\}$ for $i = 1, \dots, m$. Thus (26.35) implies

$$\nabla_i f(\bar{x}) = 0, \quad i = 1, \dots, m,$$

and this proves the assertion. \square

26.4.2 A Basic Constrained Block Descent Algorithm

Consider the constrained problem (26.2) where x is partitioned into m block components $x_i \in R^{n_i}$, for $i = 1, \dots, m$. We suppose that f is continuously differentiable on R^n and that there exists $x^0 \in X$ such that the level set

$$\mathcal{L}_X^0 = \{x \in X : f(x) \leq f(x^0)\}$$

is compact.

We recall from Chap. 4 the notion of projection of a point on a convex set and we denote by $P_{X_i}(z_i)$ the projection of a point $z_i \in R^{n_i}$ on the convex set X_i and by $P(x)$ the projection of a point $x \in R^n$ on the set X .

Under the assumptions stated, we consider the following conceptual model algorithm, which can be viewed as a block decomposition of the gradient projection method.

Block Descent Constrained (BDC) Method

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in X$, forcing functions: $\sigma_i : R^+ \rightarrow R^+$, $i = 1, \dots, m$, scale factors $a_i > 0$, $i = 1, \dots, m$.

For $k = 0, 1, \dots$

Set $z(k, 0) = x^k$. If $x^k = P[x^k - \nabla f(x^k)]$ **terminate**.

For $i = 1, \dots, m$

1. Set $d_i^k = P_{X_i}[x_i^k - a_i \nabla_i f(z(k, i-1))] - x_i^k$.

2. Set $\alpha_i^k = 0$ if $d_i^k = 0$, otherwise compute α_i^k along d_i^k using Armijo's method ALS, starting from the tentative step-size $\alpha_i = 1$.

3. Choose $x_i^{k+1} \in X_i$ in a way that the following conditions are satisfied

$$(3a) \quad f(x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k) \leq f(x_1^{k+1}, \dots, x_i^k) + \alpha_i^k d_i^k;$$

$$(3b) \quad f(x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k) \leq f(z(k, i-1)) - \sigma_i(\|x_i^{k+1} - x_i^k\|).$$

4. Set $z(k, i) = (x_1^{k+1}, \dots, x_i^{k+1}, \dots, x_m^k)$.

End For

Set $x^{k+1} = (x_1^{k+1}, \dots, x_m^{k+1})$.

End For

We note again that in the preceding scheme it is not specified the criterion used for determining the point x_i^{k+1} and we may think of some iterations of an efficient constrained minimization technique, provided that the conditions of step 3 are satisfied.

It can be easily verified that, under the assumption stated, conditions (3a) and (3b) can be satisfied, for instance, if we take x_i^{k+1} as the point obtained through Armijo's line search ALS, along the direction

$$d_i^k = P_{X_i}[x_i^k - a_i \nabla_i f(z(k, i-1))] - x_i^k = \hat{x}_i^k - x_i^k,$$

that is

$$x_i^{k+1} = x_i^k + \alpha_i^k d_i^k.$$

Indeed, from the properties of the projection operator we get

$$\begin{aligned} \nabla_i f(z(k, i-1))^T d_i^k &= \nabla_i f(z(k, i-1))^T (\hat{x}_i^k - x_i^k) \leq \\ &-1/a_i \|\hat{x}_i^k - x_i^k\|^2 = -1/a_i \|d_i^k\|^2. \end{aligned} \tag{26.37}$$

As $\alpha_i^k \leq 1$, the acceptance criterion in Armijo's method in the component space can be expressed by

$$\begin{aligned} f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) &\leq f(z(k, i-1)) + \gamma_i \alpha_i^k a_i \nabla_i f(z(k, i-1))^T d_i^k \\ &\leq f(z(k, i-1)) - \gamma_i \alpha_i^k \|d_i^k\|^2 \\ &= f(z(k, i-1)) - \sigma_i(\|(x_i^{k+1} - x_i^k)\|), \end{aligned} \tag{26.38}$$

where $\gamma_i \in (0, 1)$ and the forcing function σ_i is defined by $\sigma_i(t) = (\gamma_i)t^2$, so that conditions (3a) and (3b) are satisfied. Then, we can state the following convergence result.

Proposition 26.9 (Convergence of Block Descent Method (BDC)) Suppose that f is continuously differentiable on R^n and that there exists $x^0 \in X$ such that the level set $\mathcal{L}_X^0 = \{x \in X : f(x) \leq f(x^0)\}$ is compact. Then, the sequence $\{x^k\}$ generated by Algorithm BDC has limit points and every limit point is a critical point of f .

Proof The line search acceptance rule (26.31) implies that we can write, for $i = 1, \dots, m$:

$$f(z(k, i)) \leq f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) \leq f(z(k, i-1)), \tag{26.39}$$

which implies, as $z(k, 0) = x^k$ and $x^{k+1} = z(k, m)$, for $i = 1, \dots, m$,

$$f(x^{k+1}) \leq f(z(k, i)) \leq f(x^k). \quad (26.40)$$

As \mathcal{L}_X^0 is compact the sequence $\{f(x^k)\}$ is monotone non increasing and bounded below, so that it converges to a limit \tilde{f} . By (26.40) all the sequences $\{f(z(k, i))\}$ for $i = 0, \dots, m$ converge to the same limit, that is

$$\lim_{k \rightarrow \infty} f(z(k, i)) = \tilde{f}, \quad i = 0, \dots, m. \quad (26.41)$$

We have also that all the infinite subsequences of points produced by the method remain in \mathcal{L}_X^0 and have limit points in \mathcal{L}_X^0 . As the gradient ∇f and the projector operator are continuous and \mathcal{L}_X^0 is compact we have that the search directions $d_i^k = P_{X_i}[x_i^k - a_i \nabla_i f(z(k, i - 1))] - x_i^k$, which are feasible in X_i , are also uniformly bounded. Now, for some $i \in \{1, \dots, m\}$, suppose that the point \bar{z} is a limit point of $\{z(k, i - 1)\}$, i.e., there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{k \in K, k \rightarrow \infty} z(k, i - 1) = \bar{z}.$$

Recalling again the continuity of the gradient and of the projector operator we can write

$$\lim_{k \in K, k \rightarrow \infty} \hat{x}_i^k = \hat{x}_i = P_{X_i}[\bar{x}_i - a_i \nabla_i f(\bar{z})].$$

From (26.41) and (26.39) it follows that

$$\lim_{k \rightarrow \infty} f(z(k, i - 1)) - f(x_1^{k+1}, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) = 0.$$

Therefore, by Proposition 26.1, by identifying the sequence $\{v^k\}$ considered there with the subsequence of $\{z(k, i - 1)\}$ converging to \bar{z} , we obtain

$$\begin{aligned} \lim_{k \in K, k \rightarrow \infty} \nabla_i f(z(k, i - 1))^T d_i^k &= \lim_{k \in K, k \rightarrow \infty} \nabla_i f(z(k, i - 1))^T (\hat{x}_i^k - x_i^k) \\ &= \nabla_i f(\bar{z})^T (\hat{x}_i - \bar{x}_i) = 0, \quad i = 1, \dots, m. \end{aligned} \quad (26.42)$$

Taking the limits in (26.37) for $k \in K$ and $k \rightarrow \infty$, from (26.42) it follows

$$\hat{x}_i = P_{X_i}[\bar{x}_i - a_i \nabla_i f(\bar{z})] = \bar{x}_i. \quad (26.43)$$

We also have that (26.41) and condition (3b) imply

$$\lim_{k \rightarrow \infty} \|z(k, i) - z(k, i - 1)\| = 0, \quad i = 1, \dots, m. \quad (26.44)$$

As $x^k = z(k, 0)$, if \tilde{x} is a limit point of $\{x^k\}$, Eq. (26.44) implies, by induction, that \tilde{x} is also a limit point of $\{z(k, i - 1)\}$ for $i = 1, \dots, m$. Thus (26.43) implies

$$\tilde{x}_i = P_{X_i}[\tilde{x}_i - a_i \nabla_i f(\tilde{x})], \quad i = 1, \dots, m,$$

and this proves the assertion. \square

26.5 The Gauss-Southwell Algorithm

In the Gauss-Southwell algorithm, as in the Gauss-Seidel method, the vector of variables x is partitioned into m prefixed blocks

$$x = (x_1, \dots, x_i, \dots, x_m),$$

where $x_i \in R^{n_i}$, for $i = 1, \dots, m$. However, differently from the Gauss-Seidel method, the Gauss-Southwell updates at each iteration only one block of variables. A crucial issue concerns the rule for selecting the block to be updated at any iteration k .

The basic idea, following a greedy selection, is to select the block of variables that *mostly violates* the optimality condition. To this regard we need to recall the following well-known result. Given a point $\bar{x} \in X$, the following statements are equivalent:

- (1) \bar{x} is a critical point;
- (2) $\|\bar{x} - P_X(\bar{x} - s \nabla f(\bar{x}))\| = 0$ for all $s > 0$;
- (3) $\nabla f(\bar{x})^T(x - \bar{x}) \geq 0 \quad \forall x \in X$.

Then, at iteration k , the block of variables that “mostly violates the optimality condition” depends on the chosen condition. In the sequel we limited ourselves to the employment of optimality condition (2) (the analysis in the case of adoption of condition (3) is very similar). Then the index $i(k) \in \{1, \dots, m\}$ corresponding to the block of variables that *mostly violates* the optimality condition at iteration k is the index such that

$$\begin{aligned} & \|x_{i(k)}^k - P_{X_{i(k)}}(x_{i(k)}^k - \nabla_{i(k)} f(x^k))\| \\ & \geq \|x_j^k - P_{X_j}(x_j^k - \nabla_j f(x^k))\| \quad j = 1, \dots, m. \end{aligned} \tag{26.45}$$

Once identified the index $i(k)$, the block of variables $x_{i(k)}$ is updated as follows:

$$x_{i(k)}^{k+1} \in \operatorname{Arg} \min_{\xi \in X_{i(k)}} f(x_1^k, \dots, \xi, \dots, x_m^k).$$

Remark 26.1 Note that, given a point $x \in R^n$, the computation of the projected point \hat{x} of x on the convex set X requires to solve the problem

$$\min_y \|y - x\|^2$$

$$y \in X = X_1 \times X_2 \times \dots \times X_m,$$

which is equivalent to m independent problems

$$\min_{y_i} \|y_i - x_i\|^2$$

$$y_i \in X_i \quad i = 1, \dots, m.$$

We also observe that in the unconstrained case, i.e. $X = R^n$, condition (26.45) reduces to the condition $\|\nabla_{i(k)} f(x^k)\| \geq \|\nabla_j f(x^k)\| \quad j = 1, \dots, m$. \square

We are ready to formally define the Gauss-Southwell method.

The Gauss-Southwell Method (GSW)

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in X$.

For $k = 0, 1, \dots$

Let $i(k) \in \{1, \dots, m\}$ be the index such that (26.45) holds, and set

$$x_{i(k)}^{k+1} \in \operatorname{Arg} \min_{\xi \in X_{i(k)}} f(x_1^k, \dots, \xi, \dots, x_m^k).$$

$$\text{Set } x^{k+1} = (x_1^k, \dots, x_{i(k)}^{k+1}, \dots, x_m^k).$$

End For

Global convergence properties of the GSW method can be proved even without convexity assumptions on the objective function.

Proposition 26.10 (Convergence of Algorithm (GSW)) *Let $f : R^n \rightarrow R$ be continuously differentiable on an open set containing the closed convex set $X \subseteq R^n$ and assume that the level set $\mathcal{L}_X^0 = \{x \in X : f(x) \leq f(x^0)\}$ is compact. Then the sequence $\{x^k\}$ generated by Algorithm GSW admits limit points and each limit point is a critical point.*

Proof The instructions of the algorithm imply for all k that $x^k \in X$ and $f(x^{k+1}) \leq f(x^k)$, so that the points of the sequence $\{x^k\}$ belong to the compact set \mathcal{L}_X^0 , and then $\{x^k\}$ admits limit points.

In order to prove the thesis, by contradiction, let us assume that there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{k \in K, k \rightarrow \infty} x^k = \bar{x}, \quad \| \bar{x} - \hat{x} \| > 0, \quad (26.46)$$

where

$$\hat{x} = P_X(\bar{x} - \nabla f(\bar{x})).$$

Then there exist an index $h \in \{1, \dots, m\}$ and a scalar $\eta > 0$ such that

$$\| \bar{x}_h - \hat{x}_h \| \geq 2\eta > 0, \quad (26.47)$$

where

$$\hat{x}_h = P_{X_h}(\bar{x}_h - \nabla_h f(\bar{x})).$$

From (26.45) and (26.47), all $k \in K$ and k sufficiently large we can write

$$\| x_{i(k)}^k - \hat{x}_{i(k)}^k \| \geq \| x_h^k - \hat{x}_h^k \| \geq \eta > 0, \quad (26.48)$$

where

$$\hat{x}_h^k = P_{X_h}(x_h^k - \nabla_h f(x^k))$$

$$\hat{x}_{i(k)}^k = P_{X_{i(k)}}(x_{i(k)}^k - \nabla_{i(k)} f(x^k))$$

Since $i(k)$ belongs to the finite set $\{1, \dots, m\}$, there exists a further subset of K (relabelled K) such that

$$i(k) = i^* \quad k \in K.$$

Note that the continuity of the projection operator and of the gradient implies

$$\lim_{k \in K, k \rightarrow \infty} \hat{x}_{i^*}^k = \lim_{k \in K, k \rightarrow \infty} P_{X_{i^*}}(x_{i^*}^k - \nabla_{i^*} f(x^k)) = P_{X_{i^*}}(\bar{x}_{i^*} - \nabla_{i^*} f(\bar{x})) = \hat{x}_{i^*}.$$

For each $k \in K$, from (26.48) it follows

$$\| x_{i^*}^k - \hat{x}_{i^*}^k \| \geq \| x_h^k - \hat{x}_h^k \| \geq \eta > 0. \quad (26.49)$$

From the properties of the projection operator we get

$$\nabla_{i^*} f(x^k)^T (\hat{x}_{i^*}^k - x_{i^*}^k) \leq -\|\hat{x}_{i^*}^k - x_{i^*}^k\|^2 \leq -\eta^2 < 0. \quad (26.50)$$

Let $d_{i^*}^k = \hat{x}_{i^*}^k - x_{i^*}^k$. For $k \in K$ and $k \rightarrow \infty$ we have $\hat{x}_{i^*}^k \rightarrow \hat{x}_{i^*}$ and $x_{i^*}^k \rightarrow \bar{x}_{i^*}$, so that the sequence $\{d_{i^*}^k\}$ is bounded. Furthermore, from (26.50) we have $\nabla_{i^*} f(x^k)^T d_{i^*}^k < 0$.

Thus the sequences $\{x^k\}$ and $\{d_{i^*}^k\}$ are such that Assumption 26.1 is satisfied provided that we identify $\{v^k\}$ with $\{x^k\}$.

Now, for $k \in K$ suppose that we compute $\alpha_{i^*}^k$ by employing the line search algorithm ALS. Then we have

$$f(x_1^k, \dots, x_{i^*}^k + \alpha_{i^*}^k d_{i^*}^k, \dots, x_m^k) \leq f(x^k).$$

Since $x_{i^*}^k \in X_{i^*}$, $x_{i^*}^k + d_{i^*}^k \in X_{i^*}$ and $\alpha_{i^*}^k \in (0, 1]$, by convexity of X_{i^*} we have

$$x_{i^*}^k + \alpha_{i^*}^k d_{i^*}^k \in X_{i^*}.$$

As

$$f(x^{k+1}) = \min_{\xi_{i^*} \in X_{i^*}} f(x_1^k, \dots, \xi_{i^*}, \dots, x_m^k),$$

we can write

$$f(x^{k+1}) \leq f(x_1^k, \dots, x_{i^*}^k + \alpha_{i^*}^k d_{i^*}^k, \dots, x_m^k) \leq f(x^k) \quad (26.51)$$

The decreasing sequence $\{f(x^k)\}$ converges and hence we have

$$\lim_{k \in K, k \rightarrow \infty} f(x^k) - f(x_1^k, \dots, x_{i^*}^k + \alpha_{i^*}^k d_{i^*}^k, \dots, x_m^k) = 0.$$

Then, by Proposition 26.1, where we identify $\{v^k\}$ with $\{x^k\}_K$, we have

$$\lim_{k \in K, k \rightarrow \infty} \nabla_{i^*} f(x^k)^T (\hat{x}_{i^*}^k - x_{i^*}^k) = 0,$$

and hence, recalling (26.50), we can write

$$\lim_{k \in K, k \rightarrow \infty} \|\hat{x}_{i^*}^k - x_{i^*}^k\| = 0,$$

which contradicts (26.49). □

26.6 Decomposition with Variable and Overlapping Blocks

In the decomposition methods previously analyzed the blocks of variables are *prefixed*, kept constant during the iterates, and each variable belongs to only one block. For sake of generality we can study decomposition methods where:

- the partition of the variables may vary from one iteration to another;
- it may happen that some variables belong to different blocks in successive iterations, thus having overlapping of blocks.

Just as example, consider a function f of three variables x_1, x_2, x_3 . In a general decomposition framework we can think, for instance, that:

- at iteration k we have two blocks of variables, (x_1, x_2) and x_3 , respectively;
- at iteration $k + 1$ we have two blocks of variables, x_1 and (x_2, x_3) , respectively;
- at iteration $k + 2$ we have the two blocks of variables, (x_1, x_2) and (x_2, x_3) , respectively.

In order to take into account the above possibilities we will use a different formalism to define a general decomposition framework.

For simplicity we will refer to the unconstrained optimization problem

$$\min_{x \in R^n} f(x).$$

At each iteration k , the vector of variables x^k is partitioned into two sub-vectors $(x_{W^k}^k, x_{\overline{W}^k}^k)$, where the index set $W^k \subset \{1, \dots, n\}$ identifies the variables of the subproblem to be solved and is called *working set*, and

$$\overline{W^k} = \{1, \dots, n\} \setminus W^k$$

is the complement index set. Starting from the current solution $x^k = (x_{W^k}^k, x_{\overline{W}^k}^k)$, the sub-vector $x_{W^k}^{k+1}$ is computed by solving the subproblem

$$\min_{x_{W^k}} f(x_{W^k}, x_{\overline{W}^k}^k) \quad (26.52)$$

The sub-vector $x_{\overline{W}^k}^{k+1}$ is not updated, that is $x_{\overline{W}^k}^{k+1} = x_{\overline{W}^k}^k$, and the current solution is updated by setting

$$x^{k+1} = (x_{W^k}^{k+1}, x_{\overline{W}^k}^{k+1}).$$

We can formally define a general decomposition algorithm.

General Decomposition Algorithm (GEDA)

Initialization. Choose $x^0 \in R^n$ and set $k = 0$.

While (the stopping criterion is not satisfied)

1. Select the working set W^k ;
2. Compute a solution $x_{W^k}^*$ of problem (26.52);
3. Set $x_i^{k+1} = \begin{cases} x_i^* & \text{if } i \in W^k \\ x_i^k & \text{otherwise;} \end{cases}$
4. Set $k = k + 1$.

End While

The convergence properties of the above scheme depend on the working set selection rule. We will define two rules leading to decomposition algorithms that include, as particular cases, the Gauss-Seidel algorithm and the Gauss-Southwell algorithm previously analyzed. The first rule is a *cyclic rule*, indeed it requires that each variable x_j , with $j \in \{1, \dots, n\}$, is cyclically inserted in the working set with a period bounded above by a given integer. Formally we define the following rule.

Working Set Selection Rule 1 (WS1) There exists an integer $M > 0$ such that, for each $k \geq 0$ and for each $j \in \{1, \dots, n\}$, there exists an index $j(k)$, with $0 \leq j(k) \leq M$, such that

$$j \in W^{k+j(k)}.$$

It can be easily verified that the WS1 defines, as particular case, the Gauss-Seidel method. In order to guarantee global convergence properties, we must ensure that $\|x^{k+1} - x^k\| \rightarrow 0$ for $k \rightarrow \infty$. This can be ensured, for instance, under a (componentwise) strong quasi-convexity assumption. Before to prove the global convergence of the algorithm, we state a preliminary result whose proof is similar to that of Proposition 26.5.

Proposition 26.11 Let $f : R^n \rightarrow R$ be a continuous function over R^n , bounded below, and assume that it is a strong quasi-convex function with respect to any subset of variables. Let $\{x^k\}$ be the sequence generated by

(continued)

Proposition 26.11 (continued)

GEDA with any working set selection rule. Let $\{x^k\}_K$ be a subsequence convergent to a point \bar{x} . Then we have

$$\lim_{k \in K, k \rightarrow \infty} \|x^{k+1} - x^k\| = 0. \quad (26.53)$$

Proof The instructions of the algorithm imply $f(x^{k+1}) \leq f(x^k)$, so that, the sequence $\{f(x^k)\}$ converges being f bounded below. Then we have

$$\lim_{k \rightarrow \infty} (f(x^{k+1}) - f(x^k)) = 0. \quad (26.54)$$

By contradiction, let us assume that there exist a subsequence $\{x^k\}_K$ and a number $\beta > 0$ such that

$$\|x^{k+1} - x^k\| = \|x_{W^k}^{k+1} - x_{W^k}^k\| \geq \beta, \quad k \in K. \quad (26.55)$$

Since

$$\cup_{k=0}^{\infty} W^k \subseteq \{1, \dots, n\},$$

there exists an infinite subset $K_1 \subseteq K$ such that

$$W^k = W \quad \text{for all } k \in K_1.$$

For $k \in K_1$ set $s^k = (x^{k+1} - x^k)/\|x^{k+1} - x^k\|$ and consider a point in the line segment between x^k and x^{k+1} defined as follows

$$\tilde{x}^{k+1} = x^k + \lambda \beta s^k,$$

with $\lambda \in (0, 1)$. As $\{x^k\}_{K_1}$ converges to $\bar{x} = (\bar{x}_W, \bar{x}_{\bar{W}})$, and $\|s^k\| = 1$, there exist a subset $K_2 \subseteq K_1$, such that $\{\tilde{x}^{k+1}\}_{K_2}$ converges to a point $x^* = (x_W^*, x_{\bar{W}}^*)$, such that

$$\|\bar{x} - x^*\| = \|\bar{x}_W - x_W^*\| = \lambda \beta > 0.$$

By the assumption of strong quasi-convexity of f with respect to x_W and by definition of x^{k+1} we must have, for each $t \in (0, 1)$,

$$f(x_W^k, x_{\bar{W}}^k) > f((1-t)x_W^k + t(x_W^k + \lambda \beta s_W^k), x_{\bar{W}}^k) \geq f(x^{k+1}).$$

From (26.54), taking the limits for $k \in K_2$ and $k \rightarrow \infty$, we obtain

$$f(\bar{x}_W, \bar{x}_{\bar{W}}) = f((1-t)\bar{x}_W + tx_{\bar{W}}^*, \bar{x}_{\bar{W}}), \quad \text{for all } t \in (0, 1),$$

which contradicts the strong quasi-convexity assumption with respect to x_W . \square

We can state a global convergence result under a suitable strong quasi-convexity assumption on f .

Proposition 26.12 *Let $f : R^n \rightarrow R$ be a continuously differentiable function over R^n , and assume that it is a strong quasi-convexity function with respect to any subset of variables. Suppose that the level set \mathcal{L}_0 is compact. Let $\{x^k\}$ be the sequence generated by GEDA with the working set selection rule WS1. Then each limit point of $\{x^k\}$ is a critical point of f .*

Proof Let $K \subseteq \{0, 1, \dots\}$ be an infinite subset such that

$$\lim_{k \in K, k \rightarrow \infty} x^k = \bar{x},$$

and, by contradiction, suppose that \bar{x} is not a critical point of f , that is, there exists an index $i \in \{1, \dots, n\}$ such that

$$\frac{\partial f(\bar{x})}{\partial x_i} \neq 0. \quad (26.56)$$

The assumptions of Proposition 26.11 hold, so that $x^{k+1} \rightarrow \bar{x}$ for $k \in K$ and $k \rightarrow \infty$. By applying repeatedly Proposition 26.11 we can write

$$\lim_{k \in K, k \rightarrow \infty} x^{k+h} = \bar{x} \quad h = 1, \dots, N, \quad (26.57)$$

where N is any positive integer. The instructions of the algorithm imply

$$\nabla_{W^k} f(x_{W^k}^{k+1}, x_{\bar{W}}^{k+1}) = 0. \quad (26.58)$$

From the WS1 rule we get that for each $k \in K$ there exists an index $i(k)$, with $0 \leq i(k) \leq M$, such that $i \in W^{k+i(k)}$. Using (26.58) we obtain

$$\frac{\partial f(x^{k+i(k)+1})}{\partial x_i} = 0,$$

from which it follows, taking into account (26.57) and the continuity of the gradient,

$$\frac{\partial f(\bar{x})}{\partial x_i} = 0,$$

and this contradicts (26.56). \square

The second working set selection rule requires that at each iteration k the working set must contain the index corresponding to the variable that mostly violates the optimality condition, that is, the index corresponding to the largest component in absolute value of the gradient.

Working Set Selection Rule 2 (WS2) For each $k \geq 0$, indicated by $i(k) \in \{1, \dots, n\}$ the index such that

$$|\nabla_{i(k)} f(x^k)| \geq |\nabla_j f(x^k)| \quad j = 1, \dots, n,$$

we must have

$$i(k) \in W^k.$$

It can be easily verified that WS2 leads, as particular case, to the Gauss-Southwell method. The global convergence can be ensured even without convexity assumptions on f . The proof of the following proposition is not reported since it is very similar to that of Proposition 26.10.

Proposition 26.13 *Let $f : R^n \rightarrow R$ be a continuously differentiable function over R^n . Suppose that the level set \mathcal{L}_0 is compact. Let $\{x^k\}$ be the sequence generated by GEDA with the working set selection rule WS2. Then each limit point of $\{x^k\}$ is a critical point of f .* \square

Finally, we remark that also block descent methods for non convex objective functions can be defined by employing general decomposition schemes with suitable selection rules for the choice of the working set. An application to neural network training is cited in the bibliographic note at the end of the chapter.

26.7 The Jacobi Method

In this section we consider an unconstrained optimization problem

$$\min_{x \in R^n} f(x),$$

and we analyze a *parallel* decomposition algorithm, where the vector of variables x is partitioned into m prefixed blocks

$$x = (x_1, \dots, x_i, \dots, x_m),$$

where $x_i \in R^{n_i}$, for $i = 1, \dots, n$.

A parallel decomposition algorithm can be defined by assuming that, starting from the current point x^k , the block components $x_i, i = 1, \dots, m$, are independently updated by vectors u_i^{k+1} , thus generating the points $w(k, i)$ defined as follows

$$w(k, i) = (x_1^k, \dots, u_i^{k+1}, \dots, x_m^k), \quad i = 1, \dots, m.$$

The points $w(k, i)$ can be obtained by any unconstrained minimization algorithm. The new point x^{k+1} must be defined by the computed points $w(k, i)$ and by a suitable rule that ensures the satisfactory of some global convergence condition. For instance, x^{k+1} can be set equal to the “best” point $w(k, i^*)$, i.e., the point such that

$$f(w(k, i^*)) = \min_{1 \leq i \leq m} \{f(w(k, i))\}.$$

Thus we can define a globally convergent algorithm corresponding to the nonlinear version of the *Jacobi method*.

Modified Jacobi Method

Data: starting point $x^0 = (x_1^0, \dots, x_m^0) \in R^n$.

For $k = 0, 1, \dots$

For $i = 1, \dots, m$

1. Compute

$$u_i \in \operatorname{Arg} \min_{\xi \in R^{n_i}} f(x_1^k, \dots, \xi, \dots, x_m^k).$$

2. Set

(continued)

$$w(k, i) = (x_1^k, \dots, u_i, \dots, x_m^k).$$

End For

Let i^* be the index such that

$$f(w(k, i^*)) = \min_{1 \leq i \leq m} \{f(w(k, i))\}.$$

Set $z = (u_1, u_2, \dots, u_m)$ and compute $f(z)$.

If $f(z) \leq f(w(k, i^*))$ set $x^{k+1} = z$; otherwise set $x^{k+1} = w(k, i^*)$.

End For

We can state a global convergence result without convexity assumptions on the objective function.

Proposition 26.14 *Let $f : R^n \rightarrow R$ be a continuously differentiable function over R^n . Suppose that the level set \mathcal{L}_0 is compact. Let $\{x^k\}$ be the sequence generated by the modified Jacobi method. Then each limit point of $\{x^k\}$ is a critical point of f .*

Proof By contradiction, let us assume that there exists an infinite subset $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{k \in K, k \rightarrow \infty} x^k = \bar{x},$$

and

$$\|\nabla_i f(\bar{x})\| = \eta > 0 \quad (26.59)$$

for some $i \in \{1, \dots, m\}$. For $k \in K$ and k sufficiently large let $d_i^k = -\nabla_i f(x^k)$. The sequence $\{d_i^k\}_K$ is bounded and we have

$$\nabla_i f(x_k)^T d_i^k < 0 \quad \forall k \in K.$$

Thus the sequences $\{x^k\}_K$ and $\{d_i^k\}_K$ are such that Assumption 26.1 is satisfied provided that we identify $\{v^k\}$ with $\{x^k\}_K$. Now, for $k \in K$ suppose that we compute α_i^k by employing the line search algorithm ALS. We can write

$$f(w(k, i)) \leq f(x_1^k, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) < f(x^k) \quad (26.60)$$

We also have

$$f(x^{k+1}) \leq f(w(k, i)) < f(x^k),$$

so that, under the assumptions stated, the sequences $\{f(x^k)\}$ and $\{f(w(k, i))\}$ converge to the same limit. From (26.60) we get

$$\lim_{k \rightarrow \infty} f(x^k) - f(x_1^k, \dots, x_i^k + \alpha_i^k d_i^k, \dots, x_m^k) = 0.$$

Then, by Proposition 26.1, where we identify $\{v^k\}$ with $\{x^k\}_K$, we have

$$\lim_{k \in K, k \rightarrow \infty} \nabla_i f(x^k)^T d_i^k = -\|\nabla_i f(\bar{x})\|^2 = 0,$$

which contradicts (26.59). \square

26.8 Algorithms for Problems with a Single Linear Equality Constraint and Box Constraints

Let us consider the problem:

$$\min f(x)$$

$$a^T x = b \tag{26.61}$$

$$l \leq x \leq u,$$

where $f : R^n \rightarrow R$ is a continuously differentiable function, $a \in R^n$ is such that $a_i \neq 0$ for $i = 1, \dots, n$, $b \in R$, $l_i < u_i$ for $i = 1, \dots, n$. We suppose that the dimension n is so large that traditional optimization methods cannot be directly employed since basic operations, such as the updating of the gradient or the evaluation of the objective function, are too time consuming. There are many real applications that can be modelled by optimization problems of the form (26.61). For instance, training problem of Support Vector Machines, optimal control problems, portfolio selection problems, traffic equilibrium problems, multicommodity network flow problems are specific instances of (26.61).

The “classical” decomposition methods for nonlinear optimization previously considered, such as the the Jacobi and Gauss-Seidel algorithms, are applicable only when the feasible set is the Cartesian product of convex subsets defined in smaller subspaces. Then, the presence of the equality constraint in (26.61) implies that specific decomposition algorithms must be designed.

As already said, in a general decomposition framework, at each iteration k , the vector of variables x^k is partitioned into two sub-vectors $(x_W^k, x_{\bar{W}}^k)$, where the index set $W \subset \{1, \dots, n\}$ identifies the variables of the subproblem to be solved and is called *working set*, and $\bar{W} = \{1, \dots, n\} \setminus W$ (for notational convenience, we omit the dependence on k).

Starting from the current solution $x^k = (x_W^k, x_{\bar{W}}^k)$, which is a feasible point, the subvector x_W^{k+1} is computed as the solution of the subproblem

$$\begin{aligned} \min_{x_W} & f(x_W, x_{\bar{W}}^k) \\ & a_W^T x_W = b - a_{\bar{W}}^T x_{\bar{W}}^k \\ & l_W \leq x_W \leq u_W. \end{aligned} \quad (26.62)$$

The variables corresponding to \bar{W} are unchanged, that is, $x_{\bar{W}}^{k+1} = x_{\bar{W}}^k$, and the current solution is updated setting $x^{k+1} = (x_W^{k+1}, x_{\bar{W}}^{k+1})$.

The cardinality q of the working set, namely the dimension of the subproblem, must be *greater than or equal to* 2, due to the presence of the equality linear constraint, otherwise we would have $x^{k+1} = x^k$.

The selection rule of the working set strongly affects both the speed of the algorithm and its convergence properties.

We distinguish between:

- *Sequential Minimal Optimization (SMO) algorithms*, where the size of the working set is exactly equal to two; and
- *General Decomposition Algorithms*, where the size of the working set is strictly greater than two.

In the sequel we will mainly focus on SMO algorithms, since they are the most used algorithms to solve large-scale problems of the form (26.61).

26.9 Sequential Minimal Optimization (SMO) Algorithms

The decomposition methods usually adopted are the so-called “Sequential Minimal Optimization” (SMO) algorithms, since at each iteration they update the minimum number of variables, that is two.

The selection of the two variables to be updated must guarantee that

- (i) the new point x^{k+1} is feasible;
- (ii) if x^k is not a critical point then

$$f(x^{k+1}) < f(x^k).$$

Then, we focus on feasible (point (i)) and descent (point (ii)) directions with only two nonzero components.

Consider the feasible set (26.61), denoted by \mathcal{F} , that is

$$\mathcal{F} = \{x \in R^n : a^T x = b, l \leq x \leq u\}.$$

In correspondence to a feasible point x , we indicate the indices of active (lower and upper) box constraints as follows

$$L(x) = \{i : x_i = l\}, \quad U(x) = \{i : x_i = u\}.$$

The set of feasible directions at $x \in \mathcal{F}$ is the cone

$$D(x) = \{d \in R^n : a^T d = 0, d_i \geq 0, \text{ for all } i \in L(x), d_i \leq 0, \text{ for all } i \in U(x)\}.$$

Indeed, it is easy to see that $d \in D(x)$ implies that d is a feasible direction. Viceversa, if d is a feasible direction at $x \in \mathcal{F}$, then for t sufficiently small we must have

$$a^T(x + td) = b,$$

from which it follows

$$a^T d = 0.$$

Moreover, it must hold

$$x + td \geq l, \quad x + td \leq u,$$

which implies

$$d_i \geq 0 \quad \text{if } x_i = l_i,$$

and

$$d_i \leq 0 \quad \text{if } x_i = u_i.$$

We say that a feasible point x^* is a critical point of problem (26.61) if

$$\nabla f(x^*)^T d \geq 0 \quad \text{for all } d \in D(x^*).$$

Given a point $\bar{x} \in \mathcal{F}$, we define feasible directions d at \bar{x} with only two nonzero components d_i and d_j . To this aim, observing that we must have

$$a^T d = a_i d_i + a_j d_j = 0,$$

we set

$$d_i = \frac{1}{a_i} \quad d_j = -\frac{1}{a_j}.$$

Moreover:

- if $i \in L(\bar{x})$, i.e., $\bar{x}_i = l_i$, then we must have $d_i \geq 0$, which implies $a_i > 0$;
- if $i \in U(\bar{x})$, i.e., $\bar{x}_i = u_i$, then we must have $d_i \leq 0$, which implies $a_i < 0$;
- if $j \in L(\bar{x})$, i.e., $\bar{x}_j = l_j$, then we must have $d_j \geq 0$, which implies $a_j < 0$;
- if $j \in U(\bar{x})$, i.e., $\bar{x}_j = u_j$, then we must have $d_j \leq 0$, which implies $a_j > 0$.

Note that whenever $l_i < \bar{x}_i < u_i$, there are no constraints on the sign of d_i (and hence on the sign of a_i). Similarly, whenever $l_j < \bar{x}_j < u_j$, there are no constraints on the sign of d_j (and hence on the sign of a_j).

Then the sets L and U are partitioned in the subsets L^- , L^+ , and U^- , U^+ respectively, where

$$L^+(\bar{x}) = \{i \in L(\bar{x}) : a_i > 0\}, \quad L^-(\bar{x}) = \{i \in L(\bar{x}) : a_i < 0\}$$

$$U^+(\bar{x}) = \{i \in U(\bar{x}) : a_i > 0\}, \quad U^-(\bar{x}) = \{i \in U(\bar{x}) : a_i < 0\}.$$

Note that if:

- i belongs either to L^+ or to U^- , and
- j belongs either to L^- or to U^+ ,

then the corresponding direction $d^{i,j}$, defined as follows

$$d_h^{i,j} = \begin{cases} 1/a_h & \text{if } h = i \\ -1/a_h & \text{if } h = j \\ 0 & \text{otherwise} \end{cases}$$

is feasible at \bar{x} .

In order to formally characterize feasible directions \bar{x} with only two nonzero components we introduce the following index sets

$$\begin{aligned} R(\bar{x}) &= L^+(\bar{x}) \cup U^-(\bar{x}) \cup \{i : l_i < \bar{x}_i < u_i\} \\ S(\bar{x}) &= L^-(\bar{x}) \cup U^+(\bar{x}) \cup \{i : l_i < \bar{x}_i < u_i\}. \end{aligned} \tag{26.63}$$

Note that

$$R(\bar{x}) \cap S(\bar{x}) = \{i : 0 < \bar{x}_i < C\} \quad R(\bar{x}) \cup S(\bar{x}) = \{1, \dots, n\}.$$

The two index sets R and S allow us to state the following result on feasible directions with only two nonzero components.

Proposition 26.15 *Let \bar{x} be a feasible point and let $(i, j) \in \{1, \dots, n\}$, $i \neq j$ be a pair of indices.*

Then the direction $d^{i,j} \in R^n$ such that

$$d_h^{i,j} = \begin{cases} 1/a_i & \text{if } h = i \\ -1/a_j & \text{if } h = j \\ 0 & \text{otherwise} \end{cases}$$

is a feasible direction at \bar{x} if and only if $i \in R(\bar{x})$ and $j \in S(\bar{x})$.

Moreover, if

$$\frac{(\nabla f(\bar{x}))_i}{a_i} - \frac{(\nabla f(\bar{x}))_j}{a_j} < 0 \quad (26.64)$$

then $d^{i,j}$ is a descent direction for f at \bar{x} .

Proof Assume that $d^{i,j}$ is a feasible direction. We will show that $i \in R(\bar{x})$ and $j \in S(\bar{x})$. Assume, by contradiction, that $i \in R(\bar{x})$ and $j \notin S(\bar{x})$, that is, $j \in L^+(\bar{x}) \cup U^-(\bar{x})$.

If $j \in L^+(\bar{x})$ then $\bar{x}_j = l_j$ and $a_j > 0$, i.e., $d_j < 0$ and hence $d^{i,j}$ is not a feasible direction since

$$\bar{x}_j + t d_j < l_j \quad \text{for all } t > 0.$$

Similarly, if $j \in U^-(\bar{x})$ then $\bar{x}_j = u_j$ and $a_j < 0$, i.e., $d_j > 0$ and hence, again, $d^{i,j}$ is not a feasible direction.

Now assume that $i \in R(\bar{x})$ and $j \in S(\bar{x})$. We must prove that $d^{i,j}$ is such that

$$a^T d^{i,j} = 0 \quad \text{and} \quad d_h^{i,j} \geq 0 \quad \text{for all } h \in L(\bar{x}) \quad \text{and} \quad d_h^{i,j} \leq 0 \quad \text{for all } h \in U(\bar{x}).$$

From the definition of $d^{i,j}$ it follows

$$a^T d = a_i d_i^{i,j} + a_j d_j^{i,j} = 0.$$

Moreover, we have $i \in R(\bar{x})$, and hence, if $i \in L(\bar{x})$ then from (26.63) we get $i \in L^+(\bar{x})$, that is $d_i = 1/a_i > 0$. Similarly, we have $j \in S(\bar{x})$ and hence, if $j \in U(\bar{x})$, then $j \in U^+(\bar{x})$, that is $d_j = -1/a_j < 0$. The same conclusions can be obtained by assuming $i \in U(\bar{x})$ and $j \in L(\bar{x})$.

Finally, since f is a smooth function, the condition

$$\nabla f(\bar{x})^T d^{i,j} = \frac{(\nabla f(\bar{x}))_i}{a_i} - \frac{(\nabla f(\bar{x}))_j}{a_j} < 0$$

is sufficient to state that $d^{i,j}$ is a descent direction for f at \bar{x} . \square

The introduction of the index sets $R(x)$ and $S(x)$ allows us to state the optimality conditions in the following form, which is useful in the definition of decompositions algorithms.

The proof of the next proposition and related results can be found in the appendix to this chapter.

Proposition 26.16 *A feasible point x^* is a critical point of (26.61) if and only if*

$$\max_{i \in R(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \min_{j \in S(x^*)} \left\{ -\frac{(\nabla f(x^*))_j}{a_j} \right\}. \quad (26.65)$$

 \square

Given a feasible point \bar{x} , which is not a critical point of problem (26.61), a pair $i \in R(\bar{x})$, $j \in S(\bar{x})$ such that

$$\left\{ -\frac{(\nabla f(\bar{x}))_i}{a_i} \right\} > \left\{ -\frac{(\nabla f(\bar{x}))_j}{a_j} \right\}$$

is said to be a *violating pair*.

Given a violating pair (i, j) , let us consider the direction $d^{i,j}$ with two nonzero elements defined as follows

$$d_h^{i,j} = \begin{cases} 1/a_i & \text{if } h = i \\ -1/a_j & \text{if } h = j \\ 0 & \text{otherwise.} \end{cases}$$

It can be easily shown that $d^{i,j}$ is a feasible direction at \bar{x} and we have $\nabla f(\bar{x})^T d^{i,j} < 0$, i.e., $d^{i,j}$ is a descent direction. This implies that the selection of a violating pair of an SMO-type algorithm yields a strict decrease of the objective function. However, the use of generic violating pairs as working sets is not sufficient to guarantee convergence properties of the sequence generated by a decomposition algorithm.

A convergent SMO algorithm can be defined using as indices of the working set those corresponding to the “maximal violation” of the optimality conditions. More specifically, given again a feasible point x which is not a critical point of problem

(26.61), let us define

$$I(x) = \left\{ i : i \in \arg \max_{i \in R(x)} \left\{ -\frac{(\nabla f(x))_i}{a_i} \right\} \right\}$$

$$J(x) = \left\{ j : j \in \arg \min_{j \in S(x)} \left\{ -\frac{(\nabla f(x))_j}{a_j} \right\} \right\}$$

Taking into account the optimality conditions as stated in (26.65), a pair $i \in I(x)$, $j \in J(x)$ most violates the optimality conditions, and therefore, it is said to be a *maximal violating pair*.

An SMO-type algorithm using maximal violating pairs as working sets is usually called *most violating pair* (MVP) algorithm.

We remark that the condition on the working set selection rule, i.e., $i \in I(x^k)$, $j \in J(x^k)$, can be viewed as a *Gauss-Southwell* rule, since it is based on the maximum violation of the optimality conditions.

Now let us consider the case of convex quadratic objective function, i.e., problem of the form

$$\begin{aligned} \min f(x) &= \frac{1}{2}x^T Qx + c^T x \\ a^T x &= b \\ l \leq x &\leq u, \end{aligned} \tag{26.66}$$

where $Q \in R^{n \times n}$ is a symmetric and positive definite matrix with columns Q_h , $h = 1, \dots, n$, and $c \in R^n$.

We formally define the MVP algorithm for the class of quadratic problems (26.66). Note that the training of Support Vector Machines (SVM) leads to a problem of the form (26.66) being Q symmetric and, in general, positive semidefinite.

SMO-MVP Algorithm

Data. Starting point $x^0 = 0$ and the gradient $\nabla f(x^0) = c$.

Initialization. Set $k = 0$.

While (the stopping criterion is not satisfied)

1. select $i \in I(x^k)$, $j \in J(x^k)$, and set $W = \{i, j\}$;
2. compute analytically a solution $x_W^* = \begin{pmatrix} x_i^* & x_j^* \end{pmatrix}^T$ of problem (26.62);

(continued)

```

3. set  $x_h^{k+1} = \begin{cases} x_i^* & \text{for } h = i \\ x_j^* & \text{for } h = j \\ x_k^k & \text{otherwise;} \end{cases}$ 
4. set  $\nabla f(x^{k+1}) = \nabla f(x^k) + (x_i^{k+1} - x_i^k)Q_i + (x_j^{k+1} - x_j^k)Q_j;$ 
5. set  $k = k + 1.$ 

end while
Return  $x^* = x^k$ 

```

Note that:

- the solution of a subproblem in two variables can be analytically determined since the objective function is quadratic;
- the scheme requires to store a vector of size n (the gradient $\nabla f(x^k)$) and *to get* two columns, Q_i e Q_j , of the matrix Q .

As regards the convergence properties, we have the following result, established in [164].

Proposition 26.17 *Assume that the matrix Q is symmetric and positive definite and let $\{x^k\}$ be the sequence generated by SMO-MVP Algorithm. Then, $\{x^k\}$ admits limit points and each limit point is a solution of problem (26.66).*

Remark 26.2 In the particular case of SVM training problem, the matrix Q is a *kernel matrix*, i.e., a Gram matrix using a kernel function to evaluate the inner products in a feature space (see, e.g., [52]), which is positive semidefinite, and the above result is still valid, as proved in [165]. \square

26.10 Appendix: Proof of Proposition 26.16

We state and prove the results leading to the proof of Proposition 26.16.

Since the constraints of the problem are linear, we have that a feasible point x^* is a critical point if and only if the Karush-Kuhn-Tucker (KKT) conditions are

satisfied. Then, consider the Lagrangian function

$$L(x, \lambda, \xi, \hat{\xi}) = f(x) + \lambda(a^T x - b) - \xi^T(x - l) + \hat{\xi}^T(x - u).$$

Proposition 26.18 *A feasible point x^* is a critical point of problem (26.61) if and only if there exists a scalar λ^* such that*

$$\frac{(\nabla f(x^*))_i}{a_i} + \lambda^* \begin{cases} \geq 0 & \text{if } i \in L(x^*) \\ \leq 0 & \text{if } i \in U(x^*) \\ = 0 & \text{if } i \notin L(x^*) \cup U(x^*), \end{cases} \quad (26.67)$$

Proof Since the objective function is continuously differentiable and the constraints are linear, we have that a feasible point x^* is a critical point if and only if the Karush-Kuhn-Tucker (KKT) conditions are satisfied, i.e., there exist $\lambda^* \in R$, $\xi^*, \hat{\xi}^* \in R^n$ such that

$$\nabla f(x^*) + \lambda^* a - \xi^* + \hat{\xi}^* = 0 \quad (26.68)$$

$$(\xi^*)^T(x^* - l) = 0 \quad (26.69)$$

$$(\hat{\xi}^*)^T(x^* - u) = 0 \quad (26.70)$$

$$\xi^*, \hat{\xi}^* \geq 0. \quad (26.71)$$

We show that (26.68)–(26.71) hold if and only if (26.67) is satisfied.

(a) Assume that x^* is a feasible point and that (26.68)–(26.71) hold.

Let $x_i^* = l_i$, so that $i \in L(x^*)$. Condition (26.70) implies $\hat{\xi}_i^* = 0$ and hence, from (26.68) and (26.71), we obtain

$$(\nabla f(x^*))_i + \lambda^* a_i = \xi_i^* \geq 0.$$

Similarly, let $x_i^* = u_i$, so that $i \in U(x^*)$. Condition (26.69) implies $\xi_i^* = 0$ and hence, from (26.68) and (26.71), we obtain

$$(\nabla f(x^*))_i + \lambda^* a_i = -\hat{\xi}_i^* \leq 0.$$

Finally, let $l_i < x_i^* < u_i$, so that $i \notin L(x^*) \cup U(x^*)$. Condition (26.69) and (26.70) imply $\xi_i^*, \hat{\xi}_i^* = 0$ and hence, from (26.68) we obtain

$$(\nabla f(x^*))_i + \lambda^* a_i = 0.$$

(b) Assume that x^* is a feasible point and that (26.67) holds. For $i = 1, \dots, n$:

– if $x_i^* = l_i$ then set $\hat{\xi}_i^* = 0$ and

$$\xi_i^* = (\nabla f(x^*))_i + \lambda^* a_i;$$

– if $x_i^* = u_i$ then set $\xi_i^* = 0$ and

$$\hat{\xi}_i^* = -[(\nabla f(x^*))_i + \lambda^* a_i];$$

– if $l_i < x_i^* < u_i$ then set $\xi_i^* = 0$ and $\hat{\xi}_i^* = 0$.

It can be easily verified that (26.68)–(26.71) hold. \square

From Proposition 26.18 we get the following result.

Proposition 26.19 *A feasible point $x^* \in \mathcal{F}$ is a critical point of problem (26.61) if and only if there exists a scalar λ^* such that*

$$\begin{aligned} \lambda^* &\geq -\frac{(\nabla f(x^*))_i}{a_i} \quad \text{for all } i \in L^+(x^*) \cup U^-(x^*) \\ \lambda^* &\leq -\frac{(\nabla f(x^*))_i}{a_i} \quad \text{for all } i \in L^-(x^*) \cup U^+(x^*) \\ \lambda^* &= -\frac{(\nabla f(x^*))_i}{a_i} \quad \text{for all } i \notin L(x^*) \cup U(x^*). \end{aligned} \quad (26.72)$$

Now we can state a sufficient optimality condition.

Proposition 26.20 *Let x^* be a feasible point and assume that one of the sets $R(x^*)$, $S(x^*)$ is empty. Then x^* is a critical point of problem (26.61).*

Proof Suppose $S(x^*) = \emptyset$ (the case $R(x^*) = \emptyset$ is similar). Set

$$\lambda^* = \max_{i \in R(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} = \max_{i \in L^+(x^*) \cup U^-(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\}.$$

Conditions of (26.72) holds and hence x^* is a critical point of problem (26.61). \square

Now we can prove Proposition 26.16

Proposition 26.21 A feasible point x^* such that $R(x^*)$ and $S(x^*)$ are both not empty is a critical point of problem (26.61) if and only if

$$\max_{i \in R(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \min_{j \in S(x^*)} \left\{ -\frac{(\nabla f(x^*))_j}{a_j} \right\}. \quad (26.73)$$

Proof

- (a) Assume that x^* is a critical point.

Proposition 26.19 implies the existence of a scalar λ^* such that the pair (x^*, λ^*) satisfies condition (26.72) that can be rewritten as follows

$$\begin{aligned} \max_{i \in L^+(x^*) \cup U^-(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} &\leq \lambda^* \leq \min_{i \in L^-(x^*) \cup U^+(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \\ \lambda^* = -\frac{(\nabla f(x^*))_i}{a_i} &\text{ for all } i \notin L(x^*) \cup U(x^*). \end{aligned}$$

Using the definition of the sets $R(x^*)$ and $S(x^*)$ we obtain

$$\max_{i \in R(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \min_{j \in S(x^*)} \left\{ -\frac{(\nabla f(x^*))_j}{a_j} \right\},$$

and hence (26.73) holds.

- (b) Suppose that (26.73) holds.

We can introduce a scalar λ^* such that

$$\max_{i \in R(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \lambda^* \leq \min_{i \in S(x^*)} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\}, \quad (26.74)$$

from which we get that the inequalities of (26.72) are satisfied. On the other hand, the definition of the sets $R(x^*)$, $S(x^*)$ and the choice of λ^* (which implies that (26.74) holds) imply

$$\max_{\{i: l_i < x_i < u_i\}} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\} \leq \lambda^* \leq \min_{\{i: l_i < x_i < u_i\}} \left\{ -\frac{(\nabla f(x^*))_i}{a_i} \right\},$$

from which we get that even the equalities of (26.72) are satisfied. \square

26.11 Exercises

26.1 Consider the problem

$$\min_{x_1, x_2, x_3} f(x_1, x_2, x_3), \\ x_1 \in X_1, x_2 \in X_2, x_3 \in X_3$$

where $X_1 \subseteq R^{n_2}$, $X_2 \subseteq R^{n_2}$, $X_3 \subseteq R^{n_3}$ are non empty closed convex sets and $f : R^n \rightarrow R$ is a continuously differentiable function. Prove the convergence of the Gauss-Seidel method by assuming that f is strongly quasi-convex on $X = X_1 \times X_2 \times X_3$, with respect to $x_3 \in X_3$.

26.2 Prove Proposition 26.13.

26.3 Consider the problem

$$\min f(x)$$

$$x_1 + x_2 + \dots + x_n = 1$$

$$0 \leq x,$$

where $f : R^n \rightarrow R$ is a continuously differentiable function. Define a SMO-type algorithm using maximal violating pairs as working sets.

26.12 Notes and References

Suggested books for the study of decomposition methods are [12], [18]. In particular, [18] considers both sequential and parallel decomposition methods. Counterexamples showing that the Gauss-Seidel method may fail to converge towards stationary points have been presented in [214]. The global convergence of the Gauss-Seidel method under convexity assumptions on the objective function is proved in [262]. Proposition 26.4, established in [133], extends to the constrained case the result given in [262] for the cyclic coordinate method. Convergence results of Gauss-Seidel and Gauss-Southwell methods are stated in [132], [133], [178]. Inexact decomposition methods are presented in [29], [44], [132], while [102] is a paper studying decomposition methods within a nonmonotone framework. Decomposition methods for neural network training have been proposed in [37]. Block descent methods with a variable decomposition scheme and possibly overlapping blocks have been presented in [129]. Decomposition methods for problems with one equality constraints and box constraints have been deeply studied in the

context of machine learning, in particular, for training Support Vector Machine (SVM). Suggested references for SVM are the books [254], [140], [52]. Concerning the analysis of convergent decomposition methods for SVM training, suggested references are the following papers [47] [151], [164], [172], [103], and the survey [204].

Chapter 27

Basic Concepts of Linear Algebra and Analysis



In this chapter we report the basic definitions, concepts and results of linear algebra and analysis used in the book.

27.1 The Space R^n as Linear Space

The optimization problems considered in the book are defined on the space R^n , i.e., the vector of the variables is a vector in R^n . As well-known, the space R^n can be represented as a *linear space* (or *vector space*) with R as the associate scalar field, where the operations of *vector addition* and *multiplication by a scalar* must satisfy the axioms of linear spaces. The elements of R^n are called *points* or *vectors*.

First we recall the following definitions.

Definition 27.1 (Linear Subspace) We say that $W \subseteq R^n$ is a linear subspace of R^n if, for each $x, y \in W$ and $\alpha, \beta \in R$, we have $\alpha x + \beta y \in W$.

□

Definition 27.2 (Linear Combination) Let x_1, x_2, \dots, x_m be a *finite number* of elements of R^n . The element $x \in R^n$ defined by:

$$x = \sum_{i=1}^m \alpha_i x_i, \quad \alpha_i \in F$$

is a *linear combination* of x_1, x_2, \dots, x_m . □

Definition 27.3 (Linear Hull of a Set) Let S be a subset of \mathbb{R}^n . The *linear hull* of S or *linear span* of S is the set $\text{lin}(S)$ of all the linear combinations of elements of S \square

It can be easily shown that $\text{lin}(S)$ contains S and is a linear subspace given by the intersections of all the linear subspaces containing S .

Definition 27.4 (Linear Dependence and Independence of Vectors) Let x_1, x_2, \dots, x_m be given vectors in \mathbb{R}^n . We say that x_1, x_2, \dots, x_m are *linearly dependent* if there exists $\alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{R}$ not all zero such that

$$\sum_{i=1}^m \alpha_i x_i = 0;$$

otherwise we say that x_1, x_2, \dots, x_m are *linearly independent*.

Equivalently, we say that x_1, x_2, \dots, x_m are *linearly independent* if and only if

$$\sum_{i=1}^m \alpha_i x_i = 0 \quad \text{implies} \quad \alpha_i = 0, \quad i = 1, \dots, m.$$

A set $S \subseteq \mathbb{R}^n$ is linearly dependent if there exists a finite set of elements of S that are linearly dependent. Otherwise, we say that S is linearly independent. Equivalently, we say that S is linearly independent if *every finite set of elements of S is linearly independent*. \square

The space \mathbb{R}^n is a *finite dimensional space*, i.e., there exists a finite set of elements of \mathbb{R}^n such that each vector of \mathbb{R}^n can be expressed as a linear combination of such elements. In particular, there exists a set B , called *basis*, of n linearly independent elements such that $\mathbb{R}^n = \text{lin}(B)$.

A particular basis (called *canonical basis*) is that defined by the linearly independent vectors:

$$e_1 = (1, 0, \dots, 0), \quad e_2 = (0, 1, \dots, 0), \quad \dots \quad e_n = (0, 0, \dots, 1).$$

In fact, for every $x \in R^n$, we can write

$$x = \sum_{j=1}^n x_j e_j.$$

We state the following definitions.

Definition 27.5 (Rank of a Set) $S \subseteq R^n$. The *rank* of S is the number $r(S) \geq 0$ equal to the maximum number of linearly independent elements of S . \square

Definition 27.6 (Basis of a Set) Let $S \subseteq R^n$. The *basis* of S is a linearly independent subset $B \subseteq S$ such that the cardinality of B (number of elements of B) is equal to the rank of S , that is:

$$|B| = r(S).$$

\square

We can observe that, thanks to the introduced definitions, each subset $S \subseteq R^n$, such that $S \neq \{0\}$, always contains a basis $B \subseteq S$ composed by a finite number of elements and satisfying $|B| = r(S) \leq n$.

The following result holds.

Proposition 27.1 Let B be a basis of $S \subseteq R^n$. Then every element of S can be written in a unique way as a linear combination of elements of B . \square

27.2 Matrices and Systems of Linear Equalities

In matrix operations the point $x \in R^n$ is intended to be a *column vector* and the transpose of x , denoted by x^T , is intended to be a *row vector*.

Let A be a $m \times n$ matrix with real elements:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

We denote by A_j , $j = 1, \dots, n$ the columns of A and by a_i^T , $i = 1, \dots, m$ the rows of A . We can view the columns A_1, \dots, A_n as vectors in R^m and a_1, \dots, a_m as vectors in R^n .

If A is a $m \times n$ matrix and $x \in R^n$, the product $b = Ax$ is a vector in R^m .

We can write

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^n a_{1j}x_j \\ \vdots \\ \sum_{j=1}^n a_{mj}x_j \end{pmatrix}$$

It could be useful to consider the following representations pointing out either the rows of A :

$$\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{pmatrix}$$

or the columns of A :

$$b = \sum_{j=1}^n A_j x_j.$$

In particular, the latter representation shows that $b = Ax$ can be written as linear combination of the columns A_j of A with coefficients x_j given by the components of the vector x . In this way it is possible to refer to the columns $A_j \in R^m$ of A (viewed as elements of the linear space R^m) the concept of rank already introduced.

Definition 27.7 (Rank of a Matrix) The rank of A is the rank $r(A)$ of the set $\{A_1, A_2, \dots, A_n\} \subseteq R^m$ of the columns of A , that is the maximum number of linearly independent columns. \square

The following result is well-known and allows us to consider the concept of “rank of A ” without distinction between “row rank” and “column rank”.

Proposition 27.2 (Column Rank Equal to Row Rank) *The maximum number of linearly independent columns of A is equal to the maximum number of linearly independent rows of A , that is*

$$r(A) = r(\{A_1, \dots, A_n\}) = r(\{a_1, a_2, \dots, a_m\}).$$

□

From the preceding proposition we get that, given a $m \times n$ matrix A , we have $r(A) \leq \min\{m, n\}$. The following results can be easily established.

Proposition 27.3 *The columns of A are linearly independent, i.e., $r(A) = n$, if and only if*

$$Ax = 0 \quad \text{implies} \quad x = 0,$$

that is, if and only if the unique solution of the homogeneous system $Ax = 0$ is $x = 0$.

Proof If $Ax = 0$ then we can write:

$$0 = \sum_{j=1}^n A_j x_j$$

so that, from the definition of linearly independent vectors it follows that the columns A_j , $j = 1 \dots, n$ are linearly independent if and only if

$$\sum_{i=1}^n A_j x_j = 0 \quad \text{implies} \quad x_j = 0, \quad j = 1 \dots, n,$$

that is, if and only if $Ax = 0$ implies $x = 0$. □

Proposition 27.4 (Existence of Solutions of a Linear System) *The system $Ax = b$ admits solution if and only if $r(A) = r([A, b])$; the solution is unique if and only if $r(A) = n$.*

Proof If $Ax = b$ admits solution then

$$b = \sum_{j=1}^n A_j x_j$$

and hence b is a linear combination of the columns of A , and this implies that $r(A) = r([A, b])$. Conversely, if $r(A) = r([A, b])$ then the vector b can be written as linear combination of the columns of A , so that there exists x such that

$$b = \sum_{j=1}^n A_j x_j = Ax.$$

If there exist $x, y \in R^n$ such that $Ax = b$ and $Ay = b$ then: $A(x - y) = 0$; therefore, from Proposition 27.3 we have $x = y$ if and only if $r(A) = n$. \square

27.3 Norm, Metric, Topology, Scalar Product Over R^n

We recall the following definition formalizing the concept of “length” of a vector.

Definition 27.8 (Norm) A norm over R^n is a real function that associates at every $x \in R^n$ a real number $\|x\|$ (called *norm* of x) such that the following properties hold:

- (i) $\|x\| \geq 0$ for every $x \in R^n$;
- (ii) $\|x\| = 0$ if and only if $x = 0$;
- (iii) $\|x + y\| \leq \|x\| + \|y\|$ for every $x, y \in R^n$;
- (iv) $\|\alpha x\| = |\alpha| \|x\|$ for every $\alpha \in R, x \in R^n$.

Starting from the norm we can introduce a *metric* over R^n , that is, we can define the “distance” between two vectors $x, y \in R^n$ by setting:

$$d(x, y) = \|x - y\|.$$

The function $d : R^n \times R^n \rightarrow R$ is called *distance* or *metric*. We observe that the concept of distance can be introduced independently of that of norm and does not require an underlying linear space.

The following definition can be stated.

Definition 27.9 (Distance or Metric) Let X be a set; a function $d : X \times X \rightarrow R$ is a distance (or metric) if the following properties hold:

- (i) $d(x, y) \geq 0$ for every $x, y \in X$;
- (ii) $d(x, y) = 0$ if and only if $x = y$;
- (iii) $d(x, y) \leq d(x, z) + d(y, z)$ for every $x, y, z \in X$;
- (iv) $d(x, y) = d(y, x)$, for every $x, y \in X$.

□

It can be easily verified that, assuming $X \equiv R^n$, the function $d(x, y) = \|x - y\|$ satisfies the conditions of Definition 27.9 and hence represents a distance.

If $x \in R^n$ is a vector with components $x_1, x_2, \dots, x_n \in R$ then a norm over R^n can be defined by setting:

$$\|x\|_p = \left[\sum_{i=1}^n |x_i|^p \right]^{\frac{1}{p}}, \quad p \geq 1$$

and in this case it is called *Hölder norm* (or ℓ_p norm). In particular, for $p = 1$ we have the ℓ_1 norm:

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

and for $p = 2$ we have the *Euclidean norm* (or ℓ_2 norm):

$$\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}}.$$

Another norm in R^n (called ℓ_∞ norm or Tchebychev norm) is defined as follows:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

It can be proved that the functions $\|\cdot\|_p$ with $1 \leq p \leq \infty$ satisfy the conditions of Definition 27.8 and hence represent norms in R^n .

It can be easily verified that, using the definition of norm, the following inequality holds:

$$\|x - y\| \geq |\|x\| - \|y\||$$

from which we get that the norm is a *uniformly continuous* function in R^n .

It can be proved that in a finite dimensional space, and hence, in particular in R^n , the norms are *equivalent*, i.e., if $\|\cdot\|_a$ and $\|\cdot\|_b$ are two norms then there exist two constants $c_2 \geq c_1 > 0$ such that, for every $x \in R^n$ we have

$$c_1 \|x\|_a \leq \|x\|_b \leq c_2 \|x\|_a.$$

In particular, it can be verified that the following inequalities hold:

$$\|x\|_1 \geq \|x\|_2 \geq \|x\|_\infty \quad \|x\|_1 \leq \sqrt{n} \|x\|_2 \quad \|x\|_2 \leq \sqrt{n} \|x\|_\infty.$$

Starting from a metric and, in particular, from a metric induced by a norm, it is possible to introduce in R^n the notion of open set (and hence to define a *topology*).

Definition 27.10 (Open Ball, Closed Ball) An open ball with center x_0 and radius $\rho > 0$ is the set

$$B(x_0, \rho) = \{x \in R^n : \|x - x_0\| < \rho\};$$

a closed ball with center x_0 and radius $\rho > 0$ is the set

$$\bar{B}(x_0, \rho) = \{x \in R^n : \|x - x_0\| \leq \rho\}.$$

□

Then we can define the notion of *open set* and of *closed set*.

Definition 27.11 (Open Set, Closed Set) The set $S \subseteq R^n$ is *open* if for every $x \in S$ there exists an open ball of center x and radius ρ contained in S , that is, there exists $\rho > 0$ such that

$$B(x, \rho) \subseteq S.$$

The set $S \subseteq R^n$ is *closed* if the complement of S , that is the set $R^n/S = \{x \in R^n : x \notin S\}$, is an open set. □

Definition 27.12 (Interior and Closure of a Set) Let $S \subseteq R^n$; a point $x \in S$ is an *interior point* of S if there exists $\rho > 0$ such that $B(x, \rho) \subseteq S$.

(continued)

Definition 27.12 (continued)

A point $x \in R^n$ is a *closure point* of S if for every $\rho > 0$ it follows $B(x, \rho) \cap S \neq \emptyset$.

The *interior* of S is the set $\text{int}(S)$ of all the interior points of S ; the *closure* of S is the set \bar{S} of all the closure points of S . \square

Definition 27.13 (Boundary of a Set) Let $S \subseteq R^n$; $x \in R^n$ is a *boundary point* of S if for every $\rho > 0$ we have $B(x, \rho) \cap S \neq \emptyset$ and $B(x, \rho) \cap (R^n \setminus S) \neq \emptyset$.

We define *boundary* of S the set ∂S of the boundary points of S . \square

Taking into account the above definitions we have:

- $\text{int}(A)$ is an open set and \bar{A} is a closed set;
- A is open if and only if $A = \text{int}(A)$ and A is closed if and only if $\bar{A} = A$;
- $\bar{A} = A \cup \partial A$;
- the open ball is an open set and the closed ball is a closed set
- \emptyset and R^n are at the same time open and closed sets.

The family \mathcal{T} of the open sets is a *topology* over R^n , that is, it satisfies the following properties:

- (t₁) $\emptyset \in \mathcal{T}, R^n \in \mathcal{T}$;
- (t₂) the union of a subfamily (finite or infinite) of elements of \mathcal{T} is an element of \mathcal{T} ;
- (t₃) the intersection of a finite number of elements of \mathcal{T} is an element of \mathcal{T} .

As a consequence of the above definitions we have that:

- (t₄) the union of a finite number of closed sets is a closed set;
- (t₅) the intersection of any subfamily of closed sets is a closed set.

The introduction of a topology allows us to define the concept of limit.

Definition 27.14 (Limit of a Sequence) Let $\{x_k\}$ be a sequence of elements of R^n . We say that $\{x_k\}$ converges to $x \in R^n$ and that x is the limit of $\{x_k\}$ (we write either $\lim_{k \rightarrow \infty} x_k = x$, or $x_k \rightarrow x$) if we have:

$$\lim_{k \rightarrow \infty} \|x_k - x\| = 0,$$

(continued)

Definition 27.14 (continued)

that is, for every $\varepsilon > 0$ there exists an index k_ε such that $\|x_k - x\| < \varepsilon$ for every $k \geq k_\varepsilon$. \square

Definition 27.15 (Accumulation Point (or Limit Point) of a Sequence) We say that $x \in R^n$ is an accumulation point (or a limit point) of a sequence $\{x_k\}$ if there exists a subsequence convergent to x , that is, if for every $\varepsilon > 0$ and $m > 0$ there exists an index $k \geq m$ such that $\|x_k - x\| < \varepsilon$. \square

A subsequence $\{x_k\}$ is denoted by the symbol $\{x_k\}_K$, where K is an infinite subset of indices. Therefore, if x is an accumulation point of $\{x_k\}$, we can equivalently state that there exists a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K, k \rightarrow \infty} x_k = x.$$

Definition 27.16 (Bounded Set) A set $S \subseteq R^n$ is bounded if there exists a number $M > 0$ such that

$$\|x\| \leq M \quad \text{for every } x \in S.$$

 \square

The following result holds.

Proposition 27.5 (Accumulation Points of Bounded and of Closed Sets)

A set $S \subseteq R^n$ is bounded if and only if every sequence of elements of S admits at least an accumulation point in R^n (not necessarily in S).

A set $S \subseteq R^n$ is closed if and only if all the accumulation points of any sequence of elements of S belong to S . \square

Definition 27.17 (Compact Set) A set $S \subseteq R^n$ is compact if every sequence of elements of S admits a subsequence convergent to an element of S . \square

An equivalent characterization of compactness can be stated on finite dimensional spaces, and hence, in particular, on R^n .

Proposition 27.6 (Compact Sets in R^n) *A set $S \subseteq R^n$ is compact if and only if it is closed and bounded.* \square

By definition of compact set, we have that the following sets are compact:

- the empty set \emptyset ;
- a set with a single point $\{x\}$;
- the closed ball

$$\bar{B}(x; \rho) = \{y \in R^n : \|y - x\| \leq \rho\};$$

- the n -dimensional *box*:

$$S = \{x \in R^n : a_j \leq x_j \leq b_j, \quad j = 1, \dots, n\},$$

while the following sets are not compact

- the space R^n (it is unbounded);
- the open ball $B(x; \rho) = \{y \in R^n : \|y - x\| < \rho\}$ (it is not closed);
- the halfspace $S = \{x \in R^n : x \geq 0\}$ (it is unbounded).

The notion of *scalar product* between two vectors can be introduced in the space R^n through the following definition.

Definition 27.18 (Scalar Product (Inner Product)) A real function defined on $R^n \times R^n$ is a *scalar product* or *inner product* if in correspondence to every pair $x, y \in R^n$ it defines a number $\langle x, y \rangle$ (called product of x times y) such that the following properties hold:

- (i) $\langle x, x \rangle \geq 0$ for every $x \in R^n$;
- (ii) $\langle x, x \rangle = 0$ if and only if $x = 0$;
- (iii) $\langle x, y \rangle = \langle y, x \rangle$ for every $x, y \in R^n$;
- (iv) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle$ for every $\alpha \in R$, $x, y \in R^n$;
- (v) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ for every $x, y, z \in R^n$. \square

A scalar product over R^n (called *Euclidean scalar product*) can be defined by setting

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i$$

being x_i and y_i , for $i = 1, \dots, n$ the components of x and y respectively.

Using the notation of matrix calculus, we can write:

$$\langle x, y \rangle = x^T y = y^T x.$$

Starting from the scalar product, it is possible to introduce a norm, by setting:

$$\|x\| = \langle x, x \rangle^{\frac{1}{2}}$$

It can be verified that the function defined above satisfies the conditions in the definition of norm. In particular, the Euclidean norm can be introduced by setting:

$$\|x\|_2 = (x^T x)^{\frac{1}{2}}$$

The following inequality holds.

Proposition 27.7 (Cauchy-Schwarz Inequality) *Let $\langle x, y \rangle = x^T y$. Then we have*

$$|\langle x, y \rangle| \leq \|x\|_2 \|y\|_2$$

and the equality sign holds if and only if there exists $\alpha \in R$ such that $x = \alpha y$.

□

The notion of scalar product permits, in particular, to define geometric concepts, such as the angle between vectors and the orthogonality between vectors.

Definition 27.19 (Angle Between Two Vectors) The angle between two nonzero vectors $x, y \in R^n$ is the number $\theta \in [0, \pi]$, such that

$$\cos \theta = \frac{x^T y}{\|x\|_2 \|y\|_2}.$$

□

From the above definition we have that the angle between x and y is acute ($\theta < \frac{\pi}{2}$), right ($\theta = \frac{\pi}{2}$) or obtuse ($\theta > \frac{\pi}{2}$) depending on the fact that the scalar product $x^T y$ is positive, null or negative respectively.

Definition 27.20 (Orthogonal Vectors) Two vectors $x, y \in R^n$ are orthogonal if $x^T y = 0$. \square

27.4 Notation and Results on Real Matrices

Let A be a real matrix $m \times n$ with elements

$$a_{ij} \in R, \quad i = 1, \dots, m, j = 1, \dots, n.$$

We set

$$A = (a_{ij}).$$

27.4.1 Determinant, Minors, Trace

Let A be a real matrix $n \times n$. We indicate by $\det A$ the determinant of A . We have

$$\det(AB) = \det A \det B, \quad (\text{with } B(n \times n))$$

$$\det(A^T) = \det A,$$

$$\det(\alpha A) = \alpha^n \det A, \quad (\text{for } \alpha \in R).$$

$$\det(A^{-1}) = (\det A)^{-1} \quad (\text{if } \det A \neq 0).$$

A *submatrix* of A is a matrix obtained by deleting rows and columns of A ; a *minor* of A is the determinant of a square submatrix of A . A *principal submatrix* of A is a submatrix obtained deleting rows and columns with the same indices; a *principal minor* is the determinant of a principal submatrix.

The *trace* of A , indicated by $\text{tr } A$, is the sum of the diagonal elements of A , that is

$$\text{tr } A = \sum_{i=1}^n a_{ii}$$

and we have:

$$\operatorname{tr}(A + B) = \operatorname{tr} A + \operatorname{tr} B, \quad \operatorname{tr}(A^T) = \operatorname{tr} A,$$

$$\operatorname{tr}(\alpha A) = \alpha \operatorname{tr} A, \quad (\text{with } \alpha \in R).$$

27.4.2 Norm of a Matrix

Given a real matrix A $m \times n$, a norm of A can be introduced both considering A as a vector of $m \cdot n$ elements, and considering A as a representation of a linear operator $A : R^n \rightarrow R^m$. In the former case we can define the norm of A using any vector norm for the elements of A . However, some authors define *matrix norm* a norm satisfying also a *consistency property with respect to the product*, that is,

$$\|AB\| \leq \|A\|\|B\|.$$

In any case, the norm of a matrix satisfies the properties of a norm in the linear space of the real matrices $m \times n$, that is:

- (i) $\|A\| \geq 0$
- (ii) $\|A\| = 0$ if and only if $A = 0$;
- (iii) $\|A + B\| \leq \|A\| + \|B\|$;
- (iv) $\|\alpha A\| = |\alpha| \|A\|$.

An example of norm that can be viewed as norm of the vector of the elements is the so-called *Frobenius norm*, defined by

$$\|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2 \right)^{1/2}.$$

Note

$$\|AB\|_F \leq \|A\|_F \|B\|_F,$$

so that the consistency property is satisfied. The Frobenius norm can be also written in the form

$$\|A\|_F = \left(\operatorname{Tr}(A^T A) \right)^{1/2}.$$

If A is considered as a linear operator, a norm A can be defined by setting

$$\|A\| = \sup_{x \in R^n, x \neq 0} \frac{\|Ax\|}{\|x\|},$$

or equivalently

$$\|A\| = \sup_{\|x\|=1} \|Ax\|.$$

Assuming that the same norm is used both for x and Ax , the matrix norm so defined is called *norm induced* by the considered vector norm.

The norm induced by a vector norm satisfies the properties of the norm and even the consistency property with respect to the product.

Setting

$$\|A\|_p = \sup_{x \in R^n, x \neq 0} \frac{\|Ax\|_p}{\|x\|_p},$$

we have

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|,$$

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^n |a_{ij}|,$$

$$\|A\|_2 = (\lambda_{\max}(A^T A))^{1/2}.$$

being $\lambda_{\max}(A^T A)$ the maximum eigenvalue of $A^T A$ (which is necessarily positive provided that $A \neq 0$). If $A \neq 0$ is a real, symmetric matrix $n \times n$ we have

$$\|A\|_2 = \max_{1 \leq i \leq n} |\lambda_i(A)|,$$

being $\lambda_i = 1, \dots, n$ the eigenvalues of A . If $A \neq 0$ is symmetric positive semidefinite we have

$$\|A\|_2 = \lambda_{\max}(A).$$

The following relations hold for a $n \times n$ matrix:

$$n^{-1/2} \|A\|_F \leq \|A\|_2 \leq \|A\|_F,$$

$$n^{-1/2} \|A\|_1 \leq \|A\|_2 \leq n^{1/2} \|A\|_1,$$

$$n^{-1/2} \|A\|_\infty \leq \|A\|_2 \leq n^{1/2} \|A\|_\infty,$$

$$\max |a_{ij}| \leq \|A\|_2 \leq n \max |a_{ij}|,$$

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty},$$

$$\|AB\|_F \leq \min \{ \|A\|_2 \|B\|_F, \|A\|_F \|B\|_2 \},$$

$$\|Ax\|_2 \leq \|A\|_F \|x\|_2,$$

$$\|xy^T\|_F = \|xy^T\|_2 = \|x\|_2 \|y\|_2.$$

27.4.3 Orthogonal Matrices

A square real matrix V ($n \times n$) is *orthogonal* if the columns v_j , $j = 1, \dots, n$, are *orthonormal*, that is

$$v_h^T v_j = 0, \quad h \neq j, \quad v_j^T v_j = 1, \quad h, j = 1, \dots, n$$

so that we have

$$V^T V = I.$$

If V is orthogonal then the columns of V are linearly independent and the inverse of V is the transpose, that is $V^{-1} = V^T$. It can be easily verified that if V is orthogonal then even the rows of V are orthonormal and hence we also have $VV^T = I$. If A is a real matrix $m \times n$ and $P(m \times m)$, $Q(n \times n)$ are orthogonal matrices then we have

$$\|PAQ\|_2 = \|A\|_2 \quad \|PAQ\|_F = \|A\|_F.$$

In particular, if $Q(n \times n)$ is an orthogonal matrix then we have

$$\|Qx\|_2 = \|x\|_2.$$

27.4.4 Eigenvalues of Symmetric Matrices

We denote by Q a square, symmetric matrix $n \times n$ with real elements and we refer to the Euclidean norm $\|\cdot\|$.

As well-known, a real, symmetric, square matrix Q has n real *eigenvalues* and there exists a set of n real, non zero and orthonormal *eigenvectors*, i.e., there exist n

scalars $\lambda_i \in R$ and n vectors $u_i \in R^n$, such that

$$Qu_i = \lambda_i u_i, \quad i = 1 \dots, n,$$

$$u_i^T u_j = 0, \quad i \neq j, \quad i, j = 1 \dots, n,$$

$$\|u_i\| = 1, \quad i = 1 \dots, n.$$

Given eigenvalues and eigenvectors, a representation of Q (called *spectral representation*, or *spectral decomposition*) can be defined by setting

$$Q = \sum_{i=1}^n \lambda_i u_i u_i^T.$$

If $\{u_1, u_2, \dots, u_n\}$ is a set of orthonormal eigenvectors of Q and we introduce the orthogonal matrix $U = (u_1, \dots, u_n)$, then we have $U^T U = I$ and

$$Q = U \Lambda U^T, \quad U^T Q U = \Lambda,$$

being Λ a diagonal matrix with elements λ_i .

If $\{u_1, u_2, \dots, u_n\}$ is a set of orthonormal eigenvectors of Q then the eigenvectors are linearly independent and represent a basis of R^n . As a consequence, every $x \in R^n$ can be written in the form

$$x = \sum_{i=1}^n \alpha_i u_i$$

by a suitable choice of the coefficients α_i . It can be easily verified that, if $\|x\|$ is the Euclidean norm then

$$\|x\|^2 = \sum_{i=1}^n \alpha_i^2.$$

If $\lambda_i = 1, \dots, n$ are the eigenvalues of Q , as a consequence of a more general result concerning $n \times n$ matrices, then we have

$$\operatorname{tr} Q = \sum_{i=1}^n \lambda_i \quad \det Q = \prod_{i=1}^n \lambda_i.$$

From *Courant Fisher theorem* it follows that for any symmetric matrix $Q(n \times n)$ we have

$$\lambda_{\min}(Q) = \min_{\|x\|=1} x^T Q x = \min_{x \neq 0} \frac{x^T Q x}{\|x\|^2}, \quad (27.1)$$

$$\lambda_{\max}(Q) = \max_{\|x\|=1} x^T Q x = \max_{x \neq 0} \frac{x^T Q x}{\|x\|^2}. \quad (27.2)$$

27.4.5 Singular Value Decomposition

Let A be a real matrix $m \times n$; in order to characterize its properties in the general case it is useful to exploit the *Singular Value Decomposition* (SVD) defined in the following proposition, that can be viewed as an extension of the spectral representation valid for symmetric matrices.

Proposition 27.8 *Let A be a real matrix $m \times n$ and let $p = \min\{m, n\}$; then there exist orthogonal matrices $U(m \times m)$ and $V(n \times n)$ such that*

$$A = UDV^T,$$

being D a real matrix $(m \times n)$ such that

$$d_{ii} = \sigma_i \geq 0, i = 1, \dots, p, \quad d_{ij} = 0, i \neq j.$$

The numbers $\sigma_i, i = 1, \dots, p$ are the singular values of A . \square

Since U, V are orthogonal, we also have

$$U^T A V = D.$$

The matrix A has rank $r > 0$ if and only if there exist exactly r positive singular values, that is, we can set (possibly rearranging)

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0, \quad \sigma_{r+1} = \dots = \sigma_p = 0.$$

Setting $U = [u_1, \dots, u_m]$ e $V = [v_1, \dots, v_n]$ we have

$$AA^T u_j = \lambda_j u_j, \quad j = 1, \dots, m \quad A^T A v_j = \lambda_j v_j \quad j = 1, \dots, n$$

where

$$\lambda_j = (\sigma_j)^2, \quad j = 1, \dots, p, \quad \lambda_j = 0, \quad j = p + 1, \dots, \max\{n, m\}.$$

Therefore we have that the vectors u_j are the eigenvectors of AA^T and the vectors v_j are the eigenvectors of A^TA .

As a consequence, the singular values are the square roots of the eigenvalues either of A^TA (if $m \geq n$) or of AA^T (if $m < n$). If A is a symmetric matrix $n \times n$ then the singular values coincide with the absolute values of the eigenvalues of A .

Given the singular value decomposition, it is possible to provide the explicit representation of the minimum norm solution of a linear least squares problem, that is the solution of the problem

$$\begin{aligned} \min \quad & \|x\|_2 \\ & \|Ax - b\|_2 \leq \|Ay - b\|_2 \quad \text{for every } y \in R^n. \end{aligned} \tag{27.3}$$

Setting $A = UDV^T$ and recalling the properties of the orthogonal matrices, we can write

$$\|Ax - b\|_2 = \|UDV^Tx - b\|_2 = \|U^T(UDV^Tx - b)\|_2 = \|DV^Tx - U^Tb\|_2,$$

and similarly

$$\|Ay - b\|_2 = \|DV^Ty - U^Tb\|_2.$$

Moreover we have

$$\|x\|_2 = \|V^Tx\|_2$$

and hence, setting $z = V^Tx$ and $w = V^Ty$, problem (27.3) is equivalent to the problem

$$\begin{aligned} \min \quad & \|z\|_2 \\ & \|Dz - U^Tb\|_2 \leq \|Dw - U^Tb\|_2 \quad \text{for every } w \in R^n. \end{aligned} \tag{27.4}$$

Taking into account the definition of D , denoting by $\sigma_1, \dots, \sigma_r$ the nonzero singular values (suitably rearranged), we have that

$$\|Dw - U^Tb\|_2^2 = \sum_{i=1}^r (\sigma_i w_i - (U^Tb)_i)^2 + \sum_{i=r+1}^m (U^Tb)_i^2.$$

Then an optimal solution of the linear least squares problem must be such that

$$z_i = (U^T b)_i / \sigma_i, \quad i = 1, \dots, r.$$

Among all the vectors z satisfying such a condition, the solution minimizing the Euclidean norm of z and hence the solution of problem (27.4) is clearly the one for which $z_i = 0$, $i = r + 1, \dots, m$. It follows that, defining the matrix D^+ with elements

$$d_{ii}^+ = \begin{cases} 1/\sigma_i & \text{if } \sigma_i > 0 \\ 0 & \text{if } \sigma_i = 0, \end{cases}, \quad d_{ij}^+ = 0, \quad i \neq j$$

we have $z = D^+ U^T b$ and hence the solution of (27.3) is

$$x^* = V D^+ U^T b.$$

27.5 Quadratic Forms

We define (real) *quadratic form* a function $q : R^n \rightarrow R$ written in the form

$$q(x) = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j,$$

where q_{ij} are n^2 given real coefficients. By introducing the matrix Q ($n \times n$) with elements q_{ij} , we can set

$$q(x) = x^T Q x.$$

Without loss of generality we can assume that Q is symmetric, since

$$x^T Q x = x^T \left(\frac{Q + Q^T}{2} \right) x,$$

where $(Q + Q^T)/2$ is the *symmetric part* of Q . Let us introduce the following definitions

Definition 27.21 Let Q be a symmetric matrix $n \times n$; the matrix Q is

- | | | |
|-----------------------|----|---|
| positive definite | if | $x^T Qx > 0$ for every $x \in R^n$, $x \neq 0$; |
| positive semidefinite | if | $x^T Qx \geq 0$ for every $x \in R^n$; |
| negative definite | if | $x^T Qx < 0$ for every $x \in R^n$, $x \neq 0$; |
| negative semidefinite | if | $x^T Qx \leq 0$ for every $x \in R^n$; |
| indefinite | if | there exist $x, y \in R^n$ such that $x^T Qx > 0$
and $y^T Qy < 0$. |

□

From the definition we immediately get that *a matrix Q is negative definite (semidefinite) if and only if the matrix $-Q$ is positive definite (semidefinite) positive.* We have the following result.

Proposition 27.9 Let Q be a symmetric matrix $n \times n$. Then:

- Q is positive definite if and only if the eigenvalues of Q are positive;
- Q is positive semidefinite if and only if the eigenvalues of Q are nonnegative;
- Q is negative definite if and only if the eigenvalues of Q are negative;
- Q is negative semidefinite if and only if the eigenvalues of Q are nonpositive;
- Q is indefinite if and only if has both positive and negative eigenvalues. □

On the basis of the preceding proposition we have that in order to characterize a given symmetric matrix is sufficient to determine the eigenvalues of the matrix. However, it is possible to state some characterizations in terms of the minors of the matrix. Some necessary conditions are reported below.

Proposition 27.10 Let Q be a symmetric matrix $n \times n$. Then:

- (i) if Q is positive (negative) definite then all the diagonal elements of Q are necessarily positive (negative);
- (ii) if Q positive (negative) semidefinite then all the diagonal elements of Q are necessarily nonnegative (nonpositive);

(continued)

Proposition 27.10 (continued)

(iii) if Q is positive semidefinite or negative semidefinite and there exists an index i such that $q_{ii} = 0$, then we have

$$q_{ij} = 0 \quad j = 1, \dots, n; \quad q_{hi} = 0, \quad h = 1, \dots, n.$$

□

A necessary and sufficient condition to determine whether a symmetric matrix is *positive semidefinite* can be stated taking into account the sign of *all* the principal minors of the matrix.

Proposition 27.11 Let Q be a symmetric matrix $n \times n$. Then Q is positive semidefinite if and only if all the principal minors are nonnegative. □

In order to determine whether a symmetric matrix is *positive definite* it is sufficient to consider only the n principal minors with diagonal elements $q_{11}, q_{22}, \dots, q_{ii}$ for $i = 1, \dots, n$. The following criterion, known as *Sylvester criterion*, can be stated.

Proposition 27.12 (Sylvester Criterion) Let Q be a symmetric matrix $n \times n$ and let Δ_i , for $i = 1, \dots, n$ be the n determinants

$$\Delta_i = \det \begin{pmatrix} q_{11} & q_{12} & \cdots & q_{1i} \\ q_{21} & q_{22} & \cdots & q_{2i} \\ \cdot & \cdot & \cdots & \cdot \\ q_{i1} & q_{i2} & \cdots & q_{ii} \end{pmatrix}, \quad i = 1, \dots, n.$$

Then:

Q is positive definite if and only if $\Delta_i > 0$, $i = 1, \dots, n$

Q is negative definite if and only if $(-1)^i \Delta_i > 0$, $i = 1, \dots, n$. □

We remark that the Sylvester criterion can not be used to determine whether a matrix Q is positive semidefinite.

Sufficient conditions to state that Q is positive definite can be derived from some existing criteria on the localization of the eigenvalues. In particular, in the case of symmetric matrices the following estimates (related to the *Gerschorin conditions*)

holds:

$$\lambda_{\max}(Q) \leq \max_{1 \leq i \leq n} \left\{ q_{ii} + \sum_{j=1, j \neq i}^n |q_{ij}| \right\} \quad (27.5)$$

$$\lambda_{\min}(Q) \geq \min_{1 \leq i \leq n} \left\{ q_{ii} - \sum_{j=1, j \neq i}^n |q_{ij}| \right\}. \quad (27.6)$$

As a consequence of the last inequality we have that, by assuming that Q is a *strictly diagonal dominant* matrix, i.e.,

$$q_{ii} - \sum_{j=1, j \neq i}^n |q_{ij}| > 0, \quad \text{per ogni } i = 1, \dots, n,$$

then the matrix Q is positive definite.

Chapter 28

Differentiation in R^n



In this chapter we provide the main concepts concerning the differentiation of functions of n real variables. In particular, we state the definitions of the first and second order derivatives employed in the book and the rules for the differentiation of composite functions.

For simplicity, we assume that the functions considered are defined over all R^n . The extension to the case of functions defined on (convex) open subsets of R^n is immediate.

A useful reference text is the book [200].

28.1 First Order Derivatives of a Real Function

A vector $d \in R^n$, with $d \neq 0$ defines a *direction* in R^n . The first notion of derivative that we introduce is that of *directional derivative*.

Definition 28.1 (Directional Derivative) Let $f : R^n \rightarrow R$. We say that f admits a *directional derivative* $Df(x, d)$ at the point $x \in R^n$, along the direction $d \in R^n$, if the following limit

$$\lim_{t \rightarrow 0^+} \frac{f(x + td) - f(x)}{t} := Df(x, d)$$

exists and is finite. □

If we consider f as a function of the only variable x_j , by assuming fixed the other components of x , we can introduce the concept of *partial derivative* with respect to x_j , which correspond, for $n = 1$, to the usual concept of first order derivative.

Definition 28.2 (Partial Derivative) Let $f : R^n \rightarrow R$. We say that f admits a *partial derivative* $\partial f(x)/\partial x_j$ at the point $x \in R^n$ with respect to the variable x_j if the following limit

$$\lim_{t \rightarrow 0} \frac{f(x_1, \dots, x_j + t, \dots, x_n) - f(x_1, \dots, x_j, \dots, x_n)}{t} := \frac{\partial f(x)}{\partial x_j}$$

exists and is finite. \square

If f admits partial derivatives with respect to all the components then we will indicate by $\nabla f(x)$ the n -dimensional vector (column) of the first partial derivatives of f at x , i.e.,

$$\nabla f(x) := \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

When $n > 1$, the existence of ∇f does not imply, in general, that the value of f can be approximated, with the desired degree of accuracy, in a neighborhood of x by a linear function. On the other hand, in the study of the methods considered in this book we need that the variation of f at x with respect to an increment $d \in R^n$ of x can be *uniformly* approximated by a linear functional with respect to d . Therefore, the notion of differentiability adopted here is the *Frèchet differentiability* or *strong differentiability*, although some theoretic results may require weaker conditions.

We can state the following definition.

Definition 28.3 (Strong Differentiability) Let $f : R^n \rightarrow R$. We say that f is differentiable (*Frèchet differentiable*, or *strongly differentiable*) at $x \in R^n$ if there exists $g(x) \in R^n$ such that, for every $d \in R^n$ we have

$$\lim_{\|d\| \rightarrow 0} \frac{|f(x + d) - f(x) - g(x)^T d|}{\|d\|} = 0,$$

(continued)

Definition 28.3 (continued)

The linear functional $g(x)^T : R^n \rightarrow R$ is the (Fréchet) derivative of f at x and the (column) vector $g(x) \in R^n$ is the *gradient* of f . \square

Note that, in general, the existence of ∇f does not imply the strong differentiability. However, if $\nabla f(x)$ exists and is continuous with respect to x then f is strongly differentiable at x . The following result holds.

Proposition 28.1 *Let $f : R^n \rightarrow R$ and let $x \in R^n$. We have:*

- (i) *if f is strongly differentiable at x then f is continuous at x , $\nabla f(x)$ exists and $\nabla f(x)^T$ coincides with the Fréchet derivative of f at x ;*
- (ii) *if $\nabla f(x)$ exists and ∇f is continuous with respect to x then f is strongly differentiable at x , and the Fréchet derivative of f at x coincides with $\nabla f(x)^T$ and is continuous at x .* \square

From the preceding proposition it follows that if $\nabla f(x)$ is continuous then we can write, for every $d \in R^n$:

$$f(x + d) = f(x) + \nabla f(x)^T d + \alpha(x, d),$$

where $\alpha(x, d)$ satisfies

$$\lim_{\|d\| \rightarrow 0} \frac{\alpha(x, d)}{\|d\|} = 0.$$

If f is differentiable then it is immediate to verify that there exists the directional derivative of f along any direction $d \in R^n$ and we have $Df(x, d) \equiv \nabla f(x)^T d$.

28.2 Differentiation of a Vector Function

We introduce the following definition.

Definition 28.4 (Jacobian Matrix) Let $g : R^n \rightarrow R^m$ and $x \in R^n$. If the first partial derivatives $\partial g_i(x)/\partial x_j$ exists, for $i = 1 \dots, m$ and $j = 1 \dots, n$ at

(continued)

Definition 28.4 (continued)

x we define *Jacobian matrix* of g at x the matrix $m \times n$

$$J(x) := \begin{pmatrix} \frac{\partial g_1(x)}{\partial x_1} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m(x)}{\partial x_1} & \cdots & \frac{\partial g_m(x)}{\partial x_n} \end{pmatrix}.$$

□

The notion of differentiability can be easily extended to the case of vector functions.

Definition 28.5 (First Order Derivative of a Vector Function) Let $g : R^n \rightarrow R^m$. We say that g is (*Frèchet differentiable*, or *strongly differentiable*) at the point $x \in R^n$ if there exists a matrix $G(x)$ such that, for every $d \in R^n$ we have

$$\lim_{\|d\| \rightarrow 0} \frac{\|g(x + d) - g(x) - G(x)d\|}{\|d\|} = 0.$$

The linear operator $G(x) : R^n \rightarrow R^m$ is the Frèchet derivative of g at x . □

Even in this case, the existence of the Jacobian matrix at x does not imply differentiability of g and we have the following result.

Proposition 28.2 Let $g : R^n \rightarrow R^m$ and $x \in R^n$. We have:

- (i) if g is differentiable at x then g is continuous at x , the Jacobian matrix $J(x)$ exists and $J(x)$ coincides with the Frechét derivative of g at x ;
- (ii) if there exists the Jacobian matrix $J(x)$ of g at x and J is continuous with respect to x then g is differentiable at x , and the Frechét derivative of g at x coincides with $J(x)$ and is continuous at x . □

From the preceding proposition we get that if J is continuous then, for every $d \in R^n$, we can write:

$$g(x + d) = g(x) + J(x)d + \gamma(x, d),$$

where $\gamma(x, d)$ satisfies:

$$\lim_{\|d\| \rightarrow 0} \frac{\gamma(x, d)}{\|d\|} = 0.$$

We will also use the notation $\nabla g(x)^T$ to indicate the first derivative of g , that is

$$\nabla g(x) = J(x)^T = (\nabla g_1(x), \dots, \nabla g_m(x)).$$

Then we can set

$$g(x + d) = g(x) + \nabla g(x)^T d + \gamma(x, d).$$

28.3 Second Order Derivatives of a Real Function

Let $f : R^n \rightarrow R$ be a real function. First we recall the definition of Hessian matrix.

Definition 28.6 (Hessian Matrix) Let $f : R^n \rightarrow R$ and $x \in R^n$. If there exist second order partial derivatives $\partial^2 f(x)/\partial x_i \partial x_j$, for $i = 1, \dots, n$ and $j = 1, \dots, n$ at x , we can define the *Hessian matrix* of f at x the matrix $n \times n$

$$\nabla^2 f(x) := \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}.$$

□

Definition 28.7 (Second Order Derivative) Let $f : R^n \rightarrow R$. We say that f is twice *Frèchet* differentiable (or *twice strongly differentiable*) at the point

(continued)

Definition 28.7 (continued)

$x \in R^n$ if there exists the first derivative $\nabla f(x)^T$ (in strong sense) of f and there exists a matrix $H(x)$ ($n \times n$) such that

$$f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H(x) d + \beta(x, d),$$

where $\beta(x, d)$ satisfies:

$$\lim_{\|d\| \rightarrow 0} \frac{\beta(x, d)}{\|d\|^2} = 0.$$

The matrix $H(x)$ is called second derivative (of Fréchet) of f . □

The preceding definition shows that the existence of the second order derivative permits to uniformly approximate a function f in a neighborhood of x with the desired degree of accuracy by a quadratic function.

Even in this case the only existence of the Hessian matrix (which can be interpreted as the Jacobian matrix of $\nabla f(x)$) does not imply strong differentiability. We have the following result.

Proposition 28.3 Let $f : R^n \rightarrow R$ and $x \in R^n$. We have:

- (i) if f is twice continuously differentiable at x then the gradient ∇f exists and is continuous at x , the Hessian matrix $\nabla^2 f(x)$ exists and is a symmetric matrix and $\nabla^2 f(x)$ coincides with the second order Fréchet derivative of f at x ;
- (ii) if there exists the Hessian matrix $\nabla^2 f(x)$ at x and $\nabla^2 f$ is continuous with respect to x then f is twice continuously differentiable at x , $\nabla^2 f(x)$ is necessarily symmetric and the second order Fréchet derivative of f at x coincides with $\nabla^2 f(x)$ and is continuous at x . □

From the preceding proposition it follows that if $\nabla^2 f$ is continuous we can write for every $d \in R^n$:

$$f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x) d + \beta(x, d),$$

where $\beta(x, d)$ satisfies:

$$\lim_{\|d\| \rightarrow 0} \frac{\beta(x, d)}{\|d\|^2} = 0.$$

Under the same hypotheses we also have, as already said, that the Hessian matrix $\nabla^2 f(x)$ is a symmetric matrix, that is we have

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i}, \quad i, j = 1 \dots, n.$$

We observe that, assuming the continuity of $\nabla^2 f$, we can write

$$\nabla f(x + d) = \nabla f(x) + \nabla^2 f(x)d + \gamma(x, d),$$

where

$$\lim_{\|d\| \rightarrow 0} \frac{\gamma(x, d)}{\|d\|} = 0.$$

28.4 The Mean Value Theorem and Taylor's Theorem

In the case of differentiable functions the following results (which can also be stated under weaker assumptions) hold.

Theorem 28.1 (Mean Value Theorem) *Let $f : R^n \rightarrow R$ be a differentiable function at $x \in R^n$. Then, for every $d \in R^n$ we can write*

$$f(x + d) = f(x) + \nabla f(z)^T d,$$

where $z \in R^n$ is a suitable point (depending on x and d) such that $z = x + \zeta d$, with $\zeta \in (0, 1)$. \square

We can also state an integral formulation of this result.

Theorem 28.2 (Integral Mean Value Theorem) *Let $f : R^n \rightarrow R$ a differentiable function at $x \in R^n$. Then, for every $d \in R^n$, we can write*

$$f(x + d) = f(x) + \int_0^1 \nabla f(x + td)^T d \ dt.$$

\square

The mean value theorem can be used to provide the variation of f in correspondence to two given points x and y and we have

$$f(y) = f(x) + \nabla f(z)^T (y - z),$$

where $z = x + \xi(y - x)$, with $\xi \in (0, 1)$. Similarly we have

$$f(y) = f(x) + \int_0^1 \nabla f(x + t(y - x))^T (y - x) \, dt.$$

Using the second derivatives we have the following result.

Theorem 28.3 (Taylor's Theorem) *Let $f : R^n \rightarrow R$ be a twice continuously differentiable function at $x \in R^n$. Then, for every $d \in R^n$, we can write:*

$$f(x + d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(w) d$$

where $w \in R^n$ is a suitable point (depending on x and d) such that $w = x + \xi d$, with $\xi \in (0, 1)$. \square

Given two points x, y we have

$$f(y) = f(x) + \nabla f(x)^T (y - x) + \frac{1}{2} (y - x)^T \nabla^2 f(w) (y - x)$$

where $w = x + \xi(y - x)$, with $\xi \in (0, 1)$.

Even in this case we can consider an integral formulation.

Theorem 28.4 *Let $f : R^n \rightarrow R$ be a twice continuously differentiable function at $x \in R^n$. Then, for every $d \in R^n$ we can write:*

$$f(x + d) = f(x) + \nabla f(x)^T d + \int_0^1 (1-t)d^T \nabla^2 f(x + td) d \, dt.$$

\square

In the case of vector functions $g : R^n \rightarrow R^m$ the mean value theorem can be applied to each component g_i of g . However, there is not a result similar to that of Theorem 28.1, since the points where the derivatives of the components

are computed will be, in general, different. It is possible to consider an integral expression of the variation of g .

Theorem 28.5 *Let $g : R^n \rightarrow R^m$ a differentiable function at x . Then, for every $d \in R^n$, we can write*

$$g(x + d) = g(x) + \int_0^1 J(x + td)d \ dt,$$

where J is the Jacobian matrix of g . □

Given two points x, y , we have

$$g(y) = g(x) + \int_0^1 J(x + t(y - x))(y - x) \ dt.$$

As particular case of Theorem 28.5, if ∇f is the gradient of a twice continuously differentiable function $f : R^n \rightarrow R$, we have

$$\nabla f(x + d) = \nabla f(x) + \int_0^1 \nabla^2 f(x + td)d \ dt.$$

Therefore, given two points x, y , we can write

$$\nabla f(y) = \nabla f(x) + \int_0^1 \nabla^2 f(x + t(y - x))(y - x) \ dt.$$

28.5 Derivatives of Composite Functions

Let $g : R^n \rightarrow R^m$ and $\phi : R^m \rightarrow R^p$ be two differentiable functions. Consider the composite function $\Psi : R^n \rightarrow R^p$ defined as follows:

$$\Psi(x) = \phi(g(x)).$$

In some cases it is useful to write the derivatives of Ψ as function of the derivatives of ϕ and g . The following result holds.

Proposition 28.4 (Derivative of a Composite Function) *Let $g : R^n \rightarrow R^m$ and $\phi : R^m \rightarrow R^p$ be (strongly) differentiable functions. Then, the composite function $\Psi : R^n \rightarrow R^p$ defined by*

$$\Psi(x) = \phi(g(x))$$

is (strongly) differentiable and we have

$$\nabla \Psi(x) = \nabla g(x) \nabla \phi(y)|_{y=g(x)}$$

where the symbol $\nabla \phi(y)|_{y=g(x)}$ indicates that first the derivative of ϕ with respect to y is computed and then the replacement $y = g(x)$ is performed.

□

We observe that denoting by J, J_ϕ, J_g the Jacobian matrices of Ψ, ϕ, g with respect to the corresponding variables, we can write:

$$J(x) = J_\phi(g(x)) J_g(x).$$

28.6 Examples

In this section we will give various examples of the expression of gradients, Jacobian matrices and Hessian matrices.

Preliminarily we observe that, given a matrix $A(m \times n)$, with columns $A_j \in R^m$, $j = 1, \dots, n$ and rows $a_i^T, a_i \in R^n$, $i = 1, \dots, m$, using the dyadic representations, and pointing out the rows or the columns, we can set:

$$A = \sum_{i=1}^m e_i a_i^T, \quad e_i \in R^m,$$

$$A = \sum_{j=1}^n A_j e_j^T, \quad e_j \in R^n.$$

In particular, given $u \in R^n$ we have that $e_i u^T$ is a matrix where u is the i -th row and all the other elements are null. The matrix $u e_j^T$ has as j -th column the vector u and all the other elements are null.

If F is a vector of p differentiable functions over R^n the matrix $\nabla F(x)$ ($n \times p$) with columns $\nabla F_i(x)$ for $i = 1, \dots, p$, can be written in the following form

$$\nabla F(x) = \sum_{i=1}^p \nabla F_i(x) e_i^T, \quad e_i \in R^p. \quad (28.1)$$

In the next examples the norm $\|\cdot\|$ is the Euclidean norm.

Example 28.1 Consider the function

$$f(x) = 1/2 \|g(x)\|^2,$$

where $g : R^n \rightarrow R^m$. In this case $f(x)$ can be viewed as the composite function of $\phi(y) = 1/2\|y\|^2$ and $y = g(x)$. We have $\nabla\phi(y) = y$ and hence

$$\nabla f(x) = \nabla g(x) \nabla\phi(y)|_{y=g(x)} = \nabla g(x) g(x).$$

The same expression can be obtained setting

$$\begin{aligned} \nabla f(x) &= 1/2 \nabla \left(\sum_{i=1}^m g_i(x)^2 \right) = \sum_{i=1}^m \nabla g_i(x) g_i(x) \\ &= (\nabla g_1(x) \dots \nabla g_m(x)) \begin{pmatrix} g_1(x) \\ \vdots \\ g_m(x) \end{pmatrix} = \nabla g(x) g(x). \end{aligned}$$

Denoting by J the Jacobian matrix of g we can also write:

$$\nabla f(x) = J(x)^T g(x).$$

□

Example 28.2 Let $f : R^n \rightarrow R$ be the function defined by

$$f(x) = g(x)^T h(x),$$

where $g : R^n \rightarrow R^m$ e $h : R^m \rightarrow R$ are continuously differentiable functions. We indicate by ∇g , ∇h the transposes of the Jacobian matrices of g and h , that is

$$\nabla g(x) = (\nabla g_1(x) \dots \nabla g_m(x))$$

$$\nabla h(x) = (\nabla h_1(x) \dots \nabla h_m(x)).$$

The function f can be written in the form

$$f(x) = \sum_{j=1}^m g_j(x)h_j(x).$$

Then, using the rule for the computation of the derivative of the product of functions, we obtain

$$\begin{aligned}\nabla f(x) &= \sum_{j=1}^m \nabla g_j(x)h_j(x) + \sum_{j=1}^m \nabla h_j(x)g_j(x) \\ &= \nabla g(x)h(x) + \nabla h(x)g(x).\end{aligned}\tag{28.2}$$

□

Example 28.3 Let $f : R^n \rightarrow R$ be the function

$$f(x) = c^T x,$$

where $c \in R^n$; in this case, setting

$$g(x) = c, \quad h(x) = x,$$

we have

$$\nabla g(x) = 0, \quad \nabla h(x) = I,$$

and hence, from (28.2) it follows

$$\nabla f(x) = c.$$

□

Example 28.4 Let A be a matrix $m \times n$ with rows a_i^T , $i = 1, \dots, m$. Let $F : R^n \rightarrow R^m$ be the vector function defined as follows

$$F(x) = Ax = \begin{pmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{pmatrix}.$$

From the preceding example and from the definition of ∇F it follows

$$\nabla F = (a_1 \dots a_m) = A^T.$$

Then the Jacobian matrix J of F is

$$J(x) = \nabla F(x)^T = A.$$

Example 28.5 Let $f : R^n \rightarrow R$ be the following function

$$f(x) = \frac{1}{2}x^T Qx$$

where Q is any matrix $n \times n$. Setting

$$g(x) = \frac{1}{2}x, \quad h(x) = Qx$$

from Example 28.4 it follows

$$\nabla g(x) = \frac{1}{2}I, \quad \nabla h(x) = Q^T,$$

so that, taking into account (28.2), we obtain

$$\nabla f(x) = \frac{1}{2}Qx + \frac{1}{2}Q^Tx = \frac{1}{2}(Q + Q^T)x.$$

□

Example 28.6 Let $f : R^n \rightarrow R$ be defined as follows

$$f(x) = \frac{1}{2}x^T Qx + c^T x$$

where Q is a symmetric matrix $n \times n$. From Examples 28.3 and 28.5, as $Q = Q^T$, it follows

$$\nabla f(x) = Qx + c.$$

□

Example 28.7 Let $f : R^n \rightarrow R$ be the following function

$$f(x) = \frac{1}{2}\|Ax - b\|^2$$

where A is any matrix $m \times n$ and $b \in R^m$. Function f can be written in the following form

$$f(x) = \frac{1}{2}x^T A^T Ax - (A^T b)^T x + \frac{1}{2}\|b\|^2 = \frac{1}{2}x^T Qx + c^T x + \frac{1}{2}\|b\|^2,$$

where we set

$$Q = A^T A, \quad c = -A^T b.$$

From the preceding example we get

$$\nabla f(x) = Qx + c = A^T(Ax - b).$$

The same expression can be obtained exploiting Example 28.1, considering $f(x)$ as a composite function of $\phi(y) = \frac{1}{2}\|y\|^2$ and $y = Ax - b$, and taking into account Example 28.4.

As particular case of the considered function, we set $f(x) = \frac{1}{2}\|x\|^2$, and we have

$$\nabla f(x) = x.$$

□

Example 28.8 Let $f : R^n \rightarrow R$ be the following function

$$f(x) = \|x\|.$$

The function is continuously differentiable in the neighborhood of every $x \neq 0$. We can set

$$f(x) = [\|x\|^2]^{1/2},$$

and hence, reasoning as in the Example 28.1, we can consider f as composite function of $\phi(y) = y^{1/2}$ and $g(x) = \|x\|^2$ setting

$$f(x) = \phi(g(x)).$$

For $x \neq 0$ we have:

$$\nabla \phi(y) = \frac{1}{2y^{1/2}}, \quad \nabla \phi(y)|_{y=\|x\|^2} = \frac{1}{2[\|x\|^2]^{1/2}}, \quad \nabla g(x) = 2x.$$

Then from Proposition 28.4 we obtain for $x \neq 0$:

$$\nabla f(x) = \nabla g(x) \nabla \phi(y)|_{y=g(x)} = \frac{2x}{2[\|x\|^2]^{1/2}} = \frac{x}{\|x\|}.$$

□

Example 28.9 Let $f : R^n \rightarrow R$ be the function defined as follows

$$f(x) = \|h(x)\|,$$

where $h : R^n \rightarrow R^p$ is a continuously differentiable function. The function is continuously differentiable in the neighborhood of every x such that $h(x) \neq 0$. We can consider f as composite function of $\phi(y) = \|y\|$ and $h(x)$, setting $f(x) = \phi(h(x))$. From the preceding example, for $h(x) \neq 0$, we have

$$\nabla \phi(y) = \frac{y}{\|y\|}, \quad \nabla \phi(y)|_{y=h(x)} = \frac{h(x)}{\|h(x)\|},$$

and hence we obtain

$$\nabla f(x) = \nabla h(x) \frac{h(x)}{\|h(x)\|}, \quad h(x) \neq 0.$$

Denoting by J the Jacobian matrix of h we can write:

$$\nabla f(x) = J(x)^T \frac{h(x)}{\|h(x)\|}, \quad h(x) \neq 0.$$

□

Example 28.10 Let $u \in R^m$ be a constant vector and let $\psi : R^n \rightarrow R$ be a continuously differentiable function. Consider the vector function $F : R^n \rightarrow R^m$ defined as follows:

$$F(x) = u\psi(x) = \begin{pmatrix} u_1\psi(x) \\ \vdots \\ u_m\psi(x) \end{pmatrix}.$$

Recalling (28.1) we have

$$\nabla F(x) = \sum_{i=1}^m \nabla(u_i\psi(x))e_i^T = \sum_{i=1}^m u_i \nabla\psi(x)e_i^T = \nabla\psi(x) \sum_{i=1}^m u_i e_i^T = \nabla\psi(x)u^T.$$

□

Example 28.11 Let $u : R^n \rightarrow R^m$ a vector of continuously differentiable functions and let $\psi : R^n \rightarrow R$ a continuously differentiable function. Consider the vector function $F : R^n \rightarrow R^m$ defined as follows:

$$F(x) = u(x)\psi(x) = \begin{pmatrix} u_1(x)\psi(x) \\ \vdots \\ u_m(x)\psi(x) \end{pmatrix}.$$

We can proceed as in the preceding example and, by differentiation of the products $\psi(x)u_i(x)$, we obtain

$$\nabla F(x) = \nabla u(x)\psi(x) + \nabla\psi(x)u^T(x).$$

Example 28.12 Consider the vector function $F : R^n \rightarrow R^m$ defined in the following way:

$$F(x) = A(x)u(x),$$

where $A(x)$ is the matrix $m \times p$

$$A(x) = \begin{pmatrix} a_1^T(x) \\ \vdots \\ a_m^T(x) \end{pmatrix}$$

and the functions $a_i : R^n \rightarrow R^p$ and $u : R^n \rightarrow R^p$ are continuously differentiable. The components of $A(x)u(x)$ are $a_i(x)^T u(x)$, whose gradients, recalling (28.2), take the form

$$\nabla(a_i(x)^T u(x)) = \nabla a_i(x)u(x) + \nabla u(x)a_i(x).$$

We can proceed as in the preceding examples and, taking into account the above expression, we have

$$\begin{aligned} \nabla F(x) &= \sum_{i=1}^m \nabla(a_i(x)^T \psi(x))e_i^T = \sum_{i=1}^m (\nabla a_i(x)u(x) + \nabla u(x)a_i(x))e_i^T \\ &= \sum_{i=1}^m \nabla a_i(x)u(x)e_i^T + \nabla u(x) \sum_{i=1}^m a_i(x)e_i^T \\ &= \sum_{i=1}^m \nabla a_i(x)u(x)e_i^T + \nabla u(x)A^T(x). \end{aligned}$$

Alternatively, given $A(x) = (A_1(x) \dots A_p(x))$ where $A_j(x) \in R^m$, the function $F(x) = A(x)u(x)$ can be rewritten in the following form:

$$F(x) = \sum_{j=1}^p A_j(x)u_j(x),$$

where $u_j : R^n \rightarrow R$ is the j -th component of u . As a consequence, taking into account Example 28.11, we have:

$$\nabla F(x) = \sum_{j=1}^p \left(\nabla A_j(x) u_j(x) + \nabla u_j(x) A_j^T(x) \right) = \sum_{j=1}^p \nabla A_j(x) u_j(x) + \nabla u(x) A(x)^T.$$

□

In Tables 28.1 and 28.2 we summarize the computation of ∇f and ∇F for the functions considered in the preceding examples.

The Jacobian matrices, shown in Table 28.2, can be obtained from Table 28.1 considering them as transpose of ∇F .

In Table 28.3 we report some Hessian matrices of functions $f : R^n \rightarrow R$ that can be obtained computing the Jacobian of the gradient.

Table 28.1 Gradients

f	∇f
$1/2 \ g(x)\ ^2$	$\nabla g(x)g(x)$
$g(x)^T h(x)$	$\nabla g(x)h(x) + \nabla h(x)g(x)$
$1/2 x^T Qx + c^T x$	$1/2(Q + Q^T)x + c$
$1/2 x^T Qx + c^T x$	$Qx + c$
$Q = Q^T$	
$1/2 \ Ax - b\ ^2$	$A^T(Ax - b)$
$1/2 \ x\ ^2$	x
$\ x\ $	$x / \ x\ $
$x \neq 0$	
$\ h(x)\ $	$\nabla h(x)h(x) / \ h(x)\ $
$h(x) \neq 0$	

Table 28.2 Transposed Jacobian matrices

F	∇F
Ax	A^T
$u\psi(x)$	$\nabla\psi(x)u^T$
$u \in R^m, \psi : R^n \rightarrow R$	
$u(x)\psi(x)$	$\nabla u(x)\psi(x) + \nabla\psi(x)u(x)^T$
$u : R^n \rightarrow R^m, \psi : R^n \rightarrow R$	
$A(x)u(x)$	$\sum_{j=1}^p \nabla A_j(x)u_j(x) + \nabla u(x)A(x)^T$
$A(x) = (A_1(x) \dots A_p(x))$	
$A_j : R^n \rightarrow R^m, u : R^n \rightarrow R^p$	
$\nabla f(x)$	$\nabla^2 f(x)$
$f : R^n \rightarrow R$	

Table 28.3 Hessian matrices

f	$\nabla^2 f$
$1/2\ g(x)\ ^2$	$\nabla g(x)\nabla g(x)^T + \sum_{i=1}^m \nabla^2 g_i(x)g_i(x)$
$g : R^n \rightarrow R^m$	$\nabla g(x)\nabla h(x)^T + \nabla h(x)\nabla g(x)^T$
$g(x)^T h(x)$	$+ \sum_{i=1}^m (\nabla^2 g_i(x)h_i(x) + \nabla^2 h_i(x)g_i(x))$
$g : R^n \rightarrow R^m, h : R^n \rightarrow R^m$	
$1/2x^T Qx + c^T x$	$1/2(Q + Q^T)$
$1/2x^T Qx + c^T x$	Q
$Q = Q^T$	
$1/2\ Ax - b\ ^2$	$A^T A$
$1/2\ x\ ^2$	I

Chapter 29

Introduction to Convex Analysis



Convexity plays a major role in Optimization both in theory and in computational applications. Actually, convexity properties of optimization problems guarantee, in most of cases, that a global solution can be well characterized and that the problem can be solved efficiently. In this chapter we introduce the basic concepts of convex analysis and some elements of generalized convexity, which are used in the book.

Suggested references on convex analysis are [17, 162, 229].

29.1 Convex Sets

We start by recalling the definition of some geometric objects.

Definition 29.1 (Line) A *line* through the points $x_1, x_2 \in R^n$ is the set:

$$\begin{aligned} L &= \{x \in R^n : x = (1 - \lambda)x_1 + \lambda x_2, \quad \lambda \in R\} \\ &= \{x \in R^n : x = x_1 + \lambda(x_2 - x_1), \quad \lambda \in R\}. \end{aligned}$$

□

A line passing through $x_0 \in R^n$ and parallel to $d \in R^n$, $d \neq 0$ can be represented in the form:

$$L = \{x \in R^n : x = x_0 + \lambda d, \quad \lambda \in R\}.$$

Definition 29.2 (Ray) A *ray* passing through $x_0 \in R^n$ with direction $d \in R^n$, $d \neq 0$ is the set:

$$S = \{x \in R^n : x = x_0 + \lambda d, \lambda \in R, \lambda \geq 0\}.$$

□

Definition 29.3 (Line Segment) A *line segment* passing through the points $x_1, x_2 \in R^n$ is the set

$$[x_1, x_2] = \{x \in R^n : x = (1 - \lambda)x_1 + \lambda x_2, \lambda \in R, 0 \leq \lambda \leq 1\}$$

$$= \{x \in R^n : x = x_1 + \lambda(x_2 - x_1), \lambda \in R, 0 \leq \lambda \leq 1\}.$$

□

We can now introduce the definition of convex set.

Definition 29.4 (Convex Set) A set $C \subseteq R^n$ is *convex* if, for every pair $x_1, x_2 \in C$, the line segment $[x_1, x_2]$ is contained in C , that is

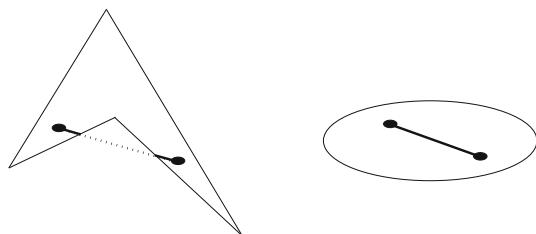
$$x_1, x_2 \in C, \lambda \in R, 0 \leq \lambda \leq 1 \text{ imply } (1 - \lambda)x_1 + \lambda x_2 \in C.$$

□

A non convex and a convex set in R^2 are shown in Fig. 29.1.

Are convex sets: the empty set, a point, a line, a ray, a line segment, an affine subspace, a linear subspace, the space R^n , a (open or closed) sphere in any norm.

Fig. 29.1 A non-convex and a convex set



It is easily seen that if $C_1 \subseteq R^n$, $C_2 \subseteq R^n$ are convex and $\alpha, \beta \in R$, then the set

$$\alpha C_1 + \beta C_2 = \{x \in R^n : x = \alpha y + \beta z, y \in C_1, z \in C_2\}$$

is convex.

The convexity of various sets can be established by employing the following result.

Proposition 29.1 (Intersection of Convex Sets) *The intersection of a family $\{C_i, i \in I\}$ (finite or infinite) of convex sets is a convex set.*

Proof Let $x_1, x_2 \in \cap_{i \in I} C_i$ and $\lambda \in R$, with $0 \leq \lambda \leq 1$. Then, for every $i \in I$ we have that $x_1, x_2 \in C_i$ and hence, by convexity of C_i , we have $(1 - \lambda)x_1 + \lambda x_2 \in C_i$. This implies

$$(1 - \lambda)x_1 + \lambda x_2 \in \cap_{i \in I} C_i$$

and hence establishes the thesis. \square

Convex sets of a special interest are the *hyperplane* and the *halfspace*.

Definition 29.5 (Hyperplane, Halfspace) Let $a \in R^n$, with $a \neq 0$ and $\beta \in R$. An *hyperplane* is the set:

$$H = \{x \in R^n : a^T x = \beta\};$$

a *closed halfspace* is the set $\{x \in R^n : a^T x \geq \beta\}$ and an *open halfspace* is the set $\{x \in R^n : a^T x > \beta\}$. \square

It can be easily verified that an hyperplane is a closed convex set and that a (closed or open) halfspace is a (closed or open) convex set.

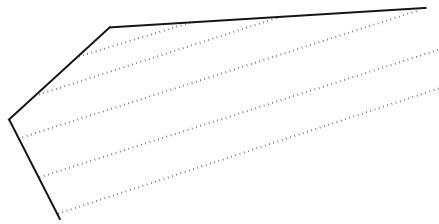
A given hyperplane in R^n determines the two closed halfspaces

$$\{x \in R^n : a^T x \geq \beta\}, \quad \{x \in R^n : a^T x \leq \beta\}$$

and it can be represented as the intersection of the two halfspaces:

$$H = \{x \in R^n : a^T x = \beta\} = \{x \in R^n : a^T x \geq \beta\} \cap \{x \in R^n : a^T x \leq \beta\}.$$

Fig. 29.2 A convex polyhedron in R^2



By Proposition 29.1, we can easily verify that the set of points that satisfy

$$a_i^T x - b_i \geq 0, \quad i = 1, \dots, m_1,$$

$$a_i^T x - b_i \leq 0, \quad i = m_1 + 1, \dots, m_2,$$

$$a_i^T x - b_i = 0, \quad i = m_2 + 1, \dots, m_3,$$

is a convex set that can be represented as the intersection of hyperplanes and closed halfspaces (or, equivalently, as intersection of closed halfspaces). We can introduce the following definition (Fig. 29.2).

Definition 29.6 (Convex Polyhedron) We define *convex polyhedron* the (closed, convex) set expressed as the intersection of a finite number of closed halfspaces. \square

Are convex polyhedra the sets

$$\{x \in R^n : Ax = b\}, \quad \{x \in R^n : Ax \geq b\},$$

$$\{x \in R^n : Ax = b, x \geq 0\}, \quad \{x \in R^n : Ax = b, Cx \geq d\}.$$

\square

Definition 29.7 (Convex Combination) Let $x_1, x_2, \dots, x_m \in R^n$. The vector $x \in R^n$ defined by

$$x = \sum_{i=1}^m \alpha_i x_i, \quad \text{with } \sum_{i=1}^m \alpha_i = 1, \quad \alpha_i \geq 0, \quad i = 1, \dots, m,$$

is said *convex combination* of the point x_1, x_2, \dots, x_m . \square

It is easily seen that the vector x is a convex combination of $x_1, x_2 \in R^n$ if and only if $x \in [x_1, x_2]$. We can establish the following proposition.

Proposition 29.2 (Necessary and Sufficient Convexity Condition) A set $C \subseteq R^n$ is convex if and only if every convex combination of elements of C belongs to C .

Proof *Sufficiency* follows directly from the definition of convex set. In fact, the line segment $[x_1, x_2]$, with $x_1, x_2 \in C$ is a (particular) convex combination of elements of C . *Necessity* can be established by induction on the number m of elements of a convex combination. For $m = 1$ and $m = 2$ the convex combinations belong to C by definition of convex set. Then, suppose that every convex combination of m elements of C belongs to C . We must show that also every convex combination of $m + 1$ elements of C belongs to C .

Let $x_1, x_2, \dots, x_m, x_{m+1} \in C$; a convex combination of these elements is given by

$$x = \sum_{i=1}^m \alpha_i x_i + \alpha_{m+1} x_{m+1}, \quad \alpha_i \geq 0, \quad i = 1, \dots, m+1, \quad \sum_{i=1}^{m+1} \alpha_i = 1.$$

If $\alpha_{m+1} = 0$ then x is a convex combination of m elements of C and hence, by the assumption made, belongs to C . If $\alpha_{m+1} = 1$ we must have

$$\sum_{i=1}^m \alpha_i = 0$$

and hence, (as $\alpha_i \geq 0$), we have $\alpha_i = 0$, for $i = 1, \dots, m$, and this implies that $x = x_{m+1} \in C$.

Finally, if $0 < \alpha_{m+1} < 1$, we have $\sum_{i=1}^m \alpha_i > 0$ and hence we can write

$$x = \sum_{i=1}^{m+1} \alpha_i x_i = \left(\sum_{i=1}^m \alpha_i \right) \left(\frac{\alpha_1 x_1}{\sum_{i=1}^m \alpha_i} + \dots + \frac{\alpha_m x_m}{\sum_{i=1}^m \alpha_i} \right) + \alpha_{m+1} x_{m+1}.$$

Now the vector

$$y = \frac{\alpha_1 x_1}{\sum_{i=1}^m \alpha_i} + \dots + \frac{\alpha_m x_m}{\sum_{i=1}^m \alpha_i}$$

is a convex combination of m elements of C . Therefore, by assumption, x belongs to C and we can write

$$x = \left(\sum_{i=1}^m \alpha_i \right) y + \alpha_{m+1} x_{m+1}, \quad y \in C, \quad x_{m+1} \in C,$$

which is a convex combination of two elements of C .

This completes the induction and establishes the thesis. \square

Now we define the concept of *extreme point*, which plays an important role in Linear Programming.

Definition 29.8 (Extreme Point) Let $C \subseteq R^n$ be a convex set. A point $x \in C$ is said to be an *extreme point* of C if it cannot be expressed as convex combination of two points of C distinct from x , that is if there not exist $y, z \in C$ with $z \neq y$ such that

$$x = (1 - \lambda)y + \lambda z, \quad \text{with } 0 < \lambda < 1.$$

\square

Are extreme points the vertices of a convex polyhedron, but not all the points of its edges; in a closed ball the extreme points are the points of the spherical surface. Convex sets like hyperplanes, halfspaces, open convex sets do not posses extreme points.

Another important definition is that of *convex cone*.

Definition 29.9 (Convex Cone) A set $K \subseteq R^n$ is said to be a *convex cone* if, for every $x, y \in K$ and $\alpha, \beta \in R$, with $\alpha \geq 0$ and $\beta \geq 0$ we have

$$\alpha x + \beta y \in K.$$

\square

It can be easily verified that a convex cone is a convex set, that the origin $x = 0 \in R^n$ belongs to a convex cone and that linear subspaces are convex cones. Are convex cones the homogeneous systems

$$\{x \in R^n : Ax = 0\}, \quad \{x \in R^n : Ax \geq 0\}.$$

The next definition introduces the concept of *convex hull* of a set $S \subseteq R^n$

Definition 29.10 (Convex Hull) Let $S \subseteq R^n$. We define *convex hull* of S the set $\text{Conv}(S)$ of all the convex combinations of elements of S . \square

We can give the following characterization of the convex hull.

Proposition 29.3 (Characterization of the Convex Hull) Let $S \subseteq R^n$. Then:

- (a) $\text{Conv}(S) \supseteq S$ and is a convex set;
- (b) $\text{Conv}(S)$ is the intersection of all convex sets containing S . \square

Proof If $x \in S$ then $1 \cdot x$ is a convex combination and hence $x \in \text{Conv}(S)$. If $x, y \in \text{Conv}(S)$ there must exist

$$x_i \in S, i = 1, \dots, m \quad \text{and} \quad y_i \in S, i = 1, \dots, p$$

such that

$$x = \sum_{i=1}^m \alpha_i x_i, \quad y = \sum_{i=1}^p \beta_i y_i,$$

with

$$\sum_{i=1}^m \alpha_i = 1, \quad \sum_{i=1}^p \beta_i = 1, \quad \alpha_i \geq 0, \quad i = 1, \dots, m, \quad \beta_i \geq 0, \quad i = 1, \dots, p.$$

Now, if $\lambda \in R$ satisfies $0 \leq \lambda \leq 1$ we have also

$$(1 - \lambda)x + \lambda y = \sum_{i=1}^m (1 - \lambda)\alpha_i x_i + \sum_{i=1}^p \lambda\beta_i y_i.$$

Moreover, we have

$$(1 - \lambda)\alpha_i \geq 0, \quad i = 1, \dots, m, \quad \lambda\beta_i \geq 0, \quad i = 1, \dots, p$$

and we can write

$$\sum_{i=1}^m (1-\lambda)\alpha_i + \sum_{i=1}^p \lambda\beta_i = (1-\lambda)\sum_{i=1}^m \alpha_i + \lambda\sum_{i=1}^p \beta_i = 1 - \lambda + \lambda = 1,$$

which implies

$$(1-\lambda)x + \lambda y \in \text{Conv}(S),$$

so that $\text{Conv}(S)$ is a convex set and (a) is proved.

Now, if C is a convex set containing S , by Proposition 29.2 every convex combination of elements of S belongs to C and hence

$$\text{Conv}(S) \subseteq C,$$

which establishes (b). \square

It is easily seen that $\text{Conv}(\{x_1, x_2\}) = [x_1, x_2]$. From the preceding results it follows that a set C is convex if and only if $C = \text{Conv}(C)$.

The next result is the *Carathéodory's Theorem*.

Proposition 29.4 (Carathéodory's Theorem) *Let $X \subseteq \mathbb{R}^n$. Then, every point $x \in \text{Conv}(X)$ can be represented as convex combination of m points of X with $m \leq n + 1$.*

Proof Let $x \in \text{Conv}(X)$, and let m be the minimal number of points of a convex combination equal to x . Then we have

$$x = \sum_{i=1}^m \alpha_i x_i \quad \sum_{i=1}^m \alpha_i = 1 \quad \alpha_i > 0 \quad i = 1, \dots, m, \quad (29.1)$$

where the strict inequality follows from the assumption made on m . Reasoning by contradiction, let us assume that $m > n + 1$ and consider the $m - 1$ vectors

$$x_2 - x_1, \quad x_3 - x_1, \quad \dots \quad x_m - x_1.$$

As $m - 1 > n$ these vectors are linearly dependent. Therefore there exist $m - 1$ numbers $\lambda_2, \dots, \lambda_m$, such that

$$\sum_{i=2}^m \lambda_i (x_i - x_1) = 0,$$

where at least one of the scalars $\lambda_i, i = 2, \dots, m$ can be chosen as positive. Letting

$$\mu_i = \lambda_i \quad \text{for } i = 2, \dots, m \quad \mu_1 = -\sum_{i=2}^m \lambda_i$$

we get

$$\sum_{i=1}^m \mu_i x_i = 0 \quad \sum_{i=1}^m \mu_i = 0, \quad (29.2)$$

where at least one of the numbers μ_2, \dots, μ_m is positive.

Let

$$\gamma = \min_{i=1, \dots, m} \left\{ \frac{\alpha_i}{\mu_i} : \mu_i > 0 \right\}$$

and define

$$\bar{\alpha}_i = \alpha_i - \gamma \mu_i \quad i = 1, \dots, m. \quad (29.3)$$

By definition of γ we have

$$\bar{\alpha}_i \geq 0 \quad i = 1, \dots, m,$$

and $\bar{\alpha}_h = 0$ for at least one index $h \in \{1, \dots, m\}$. From (29.1), (29.2) and (29.3) we obtain

$$x = \sum_{i=1}^m \bar{\alpha}_i x_i = \sum_{i=1, i \neq h}^m \bar{\alpha}_i x_i \quad \sum_{i=1}^m \bar{\alpha}_i = \sum_{i=1, i \neq h}^m \bar{\alpha}_i = 1.$$

Therefore, x can be obtained as a convex combination of a number of elements in X inferior to m , but this contradicts the definition of m . \square

29.2 Convex Functions

First we introduce some basic definitions.

Definition 29.11 (Convex Function) Let $C \subseteq R^n$ be a convex set and let $f : C \rightarrow R$. We say that f is *convex* on C if, for every pair $x, y \in C$ we have

(continued)

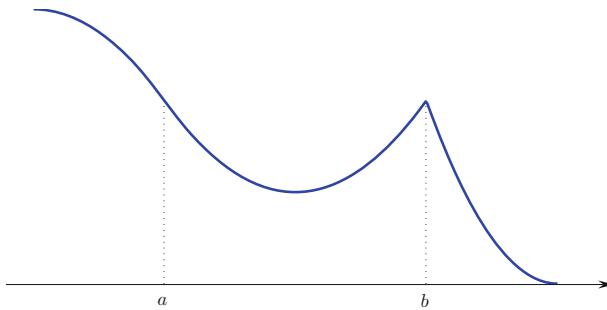


Fig. 29.3 Real function convex on $C = [a, b]$

Definition 29.11 (continued)

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y),$$

for all $\lambda \in R$ such that $0 \leq \lambda \leq 1$. We say that f is *strictly convex* on C if, for every pair $x, y \in C$ with $x \neq y$ we have

$$f((1 - \lambda)x + \lambda y) < (1 - \lambda)f(x) + \lambda f(y),$$

for all λ such that $0 < \lambda < 1$. □

In Fig. 29.3 we show a function $f : R \rightarrow R$, which is convex on the interval $C = [a, b]$ of R . In the example, f is also strictly convex on C .

From the above definitions we get immediately that of (*strictly*) *concave function* by defining f (*strictly*) concave on a convex set C if $-f$ is (*strictly*) convex on C . More explicitly, we can state the following definition.

Definition 29.12 (Concave Function) Let $C \subseteq R^n$ be a convex set and let $f : C \rightarrow R$. We say that f is *concave* on C if, for every pair $x, y \in C$ we have

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

for all $\lambda \in R$ such that $0 \leq \lambda \leq 1$. We say that f is *strictly concave* on C if, for every pair $x, y \in C$ with $x \neq y$ we have

$$f((1 - \lambda)x + \lambda y) > (1 - \lambda)f(x) + \lambda f(y),$$

(continued)

Definition 29.12 (continued)

for all λ such that $0 < \lambda < 1$. □

We can easily see that an *affine function* $f(x) = c^T x + d$ for $x \in R^n$ is both a convex and a concave function on R^n .

From the convexity of functions we can derive convexity properties of the sets defined through systems of inequalities. We show first that level sets of convex functions are convex.

Proposition 29.5 (Convexity of Level Sets) *Let $C \subseteq R^n$ be a convex set and let $f : C \rightarrow R$ a convex function on C . Then, for every $\alpha \in R$ the set $\mathcal{L}(\alpha) = \{x \in C : f(x) \leq \alpha\}$ is convex.*

Proof Let $\alpha \in R$. If $\mathcal{L}(\alpha) = \emptyset$ the assertion is true, as the empty set is convex. Then we can assume that $\mathcal{L}(\alpha)$ is nonempty. Let $x, y \in \mathcal{L}(\alpha)$ and $\lambda \in [0, 1]$. As $\mathcal{L}(\alpha) \subseteq C$ and C is convex, the point $(1 - \lambda)x + \lambda y$ belongs to C . Moreover, as f is convex, we have

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y) \leq (1 - \lambda)\alpha + \lambda\alpha = \alpha.$$

This implies

$$(1 - \lambda)x + \lambda y \in \mathcal{L}(\alpha),$$

and hence shows the convexity of $\mathcal{L}(\alpha)$. □

As the intersection of convex sets is convex, it follows from the preceding results that a *sufficient* (but not necessary) condition for the convexity of the set

$$S = \{x \in R^n : g(x) \leq 0, h(x) = 0\}$$

where $g : R^n \rightarrow R^m$ and $h : R^n \rightarrow R^p$, is that *the functions g_i are convex and the functions h_i are affine*. Note that if a set is defined as $S = \{x \in R^n : g_i(x) \geq 0\}$ a sufficient convexity condition is that g_i is concave.

We remark that the convexity of the level sets $\mathcal{L}(\alpha)$ is a *necessary but not sufficient* condition for convexity of f . A necessary and sufficient convexity condition is given in the next proposition, whose proof is left as an exercise.

Proposition 29.6 (Convexity of the Epigraph) *Let C be a convex set and let $f : C \rightarrow R$ be a function defined on C . Then, a necessary and sufficient condition for convexity of f on C is that the epigraph of f on C , defined by $\text{epi}(f) = \{(x, \alpha) \in C \times R : \alpha \geq f(x)\}$, is a convex set.* \square

29.3 Composition of Convex Functions

We give here some properties of composition of convex functions. The first case we consider is the nonnegative combination of convex functions.

Proposition 29.7 (Nonnegative Combination) *Let $C \subseteq R^n$ be a convex set and let $f_i : C \rightarrow R$, for $i = 1, \dots, m$, be convex functions on C . Then the function*

$$f(x) = \sum_{i=1}^m \alpha_i f_i(x),$$

with $\alpha_i \geq 0$, $i = 1, \dots, m$ is a convex function on C . Moreover, if, in addition, there exists at least one index i such that $\alpha_i > 0$ and f_i is strictly convex on C , then also f is strictly convex on C .

Proof Let $x, y \in C$ and $0 \leq \lambda \leq 1$. As f is convex for every i we have for $i = 1, \dots, m$:

$$f_i((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f_i(x) + \lambda f_i(y).$$

Since $\alpha_i \geq 0$ for all i , we have:

$$\begin{aligned} f((1 - \lambda)x + \lambda y) &= \sum_{i=1}^m \alpha_i f_i((1 - \lambda)x + \lambda y) \leq \sum_{i=1}^m (\alpha_i(1 - \lambda)f_i(x) + \alpha_i\lambda f_i(y)) \\ &= (1 - \lambda) \sum_{i=1}^m \alpha_i f_i(x) + \lambda \sum_{i=1}^m \alpha_i f_i(y) = (1 - \lambda)f(x) + \lambda f(y), \end{aligned}$$

which implies that f is convex.

If, in addition, we have, for at least one i , that $\alpha_i > 0$ and that f_i is strictly convex, the preceding inequality is strict for all $x, y \in C$, with $x \neq y$ and hence f is strictly convex on C . \square

A second important case is the maximum of convex functions.

Proposition 29.8 (Maximum of Convex Functions) *Let $C \subseteq R^n$ be a convex set and let $f_i : C \rightarrow R$, for $i = 1, \dots, m$ convex functions on C . Then the function*

$$f(x) = \max_{1 \leq i \leq m} \{f_i(x)\}$$

is a convex function on C . Moreover, if the functions f_i for $i = 1, \dots, m$ are strictly convex, also the function f is strictly convex.

Proof Let $x, y \in C$ and $z = (1 - \lambda)x + \lambda y$ with $0 \leq \lambda \leq 1$. We can write:

$$\begin{aligned} f(z) &= \max_{1 \leq i \leq m} \{f_i(z)\} \leq \max_{1 \leq i \leq m} \{(1 - \lambda)f_i(x) + \lambda f_i(y)\} \\ &\leq (1 - \lambda) \max_{1 \leq i \leq m} \{f_i(x)\} + \lambda \max_{1 \leq i \leq m} \{f_i(y)\} = (1 - \lambda)f(x) + \lambda f(y), \end{aligned}$$

which implies that f is convex. Moreover, if all the functions f_i are strictly convex and we have $x \neq y$ and $0 < \lambda < 1$, the first inequality in the above formula is strict and hence f is strictly convex. \square

From the preceding proposition it follows also that, if the functions $f_i : C \rightarrow R$, $i = 1, \dots, m$ are concave, then the function defined by $f(x) = \min_{1 \leq i \leq m} \{f_i(x)\}$ is concave. Note also that, in general, the minimum of convex functions may be not a convex function and that the maximum of concave functions may be not a concave function.

Another interesting result on composition of convex functions is that given in the next proposition.

Proposition 29.9 (Composite Function) *Let $C \subseteq R^n$ be a convex set, let $g : C \rightarrow R$ be a convex function and let ψ be a non decreasing convex function defined on the convex hull of the image of C , that is $\psi : \text{Conv}(g(C)) \rightarrow R$ where $g(C) = \{\alpha \in R : \alpha = g(x), x \in C\}$. Then, the composite function: $f(x) = \psi[g(x)]$ is a convex function on C . Moreover, if g is strictly convex*

(continued)

Proposition 29.9 (continued)

on C and ψ is an increasing function, strictly convex on $\text{Conv}(g(C))$, the function f is strictly convex on C .

Proof Let $x, y \in C$ and $\lambda \in [0, 1]$. Then, by convexity of g we have

$$g((1 - \lambda)x + \lambda y) \leq (1 - \lambda)g(x) + \lambda g(y),$$

and hence, as ψ is non decreasing, we can write

$$\psi[g((1 - \lambda)x + \lambda y)] \leq \psi[(1 - \lambda)g(x) + \lambda g(y)].$$

It follows, by convexity of ψ , that

$$\psi[g((1 - \lambda)x + \lambda y)] \leq \psi[(1 - \lambda)g(x) + \lambda g(y)] \leq (1 - \lambda)\psi[g(x)] + \lambda\psi[g(y)],$$

which proves the convexity of f . Moreover, if g is strictly convex on C , ψ is increasing and strictly convex on $g(C)$ and $0 < \lambda < 1$, all inequalities in the above formula are strict for $x \neq y$ and hence f is strictly convex on C . \square

In order to illustrate the preceding results, let us assume that $\phi : R^n \rightarrow R$ is a convex function and define $f : R^n \rightarrow R$ by assuming

$$f(x) = (\max\{0, \phi(x)\})^2.$$

The function $g(x) = \max\{0, \phi(x)\}$ is a convex function (as maximum of two convex functions) and it is nonnegative on R^n , so that $g(R^n) \subseteq R^+$. Then, we can view f as composition of the function $\psi(t) = t^2$ with the function $g(x) = \max\{0, \phi(x)\}$. The function $\psi(t)$ is a convex function, non decreasing for nonnegative values of t and hence, by Proposition 29.9, f is a convex function. Note that a function of the form $f(x) = \phi(x)^2$, with ϕ convex may be not convex, if ϕ takes negative values in C , because the function $\psi(t) = t^2$ is decreasing for negative values of t .

In Fig. 29.4 we have assumed $\phi(x) = x^2 - 1$ and we have considered the two cases

$$g(x) = \max\{0, \phi(x)\}, \quad g(x) = \phi(x),$$

where, in both cases we have

$$f(x) = \psi(g(x)), \quad \psi(t) = t^2.$$

A very interesting result is that convexity imply continuity on open sets. We state the following proposition, whose proof can be found, for instance, in [179].

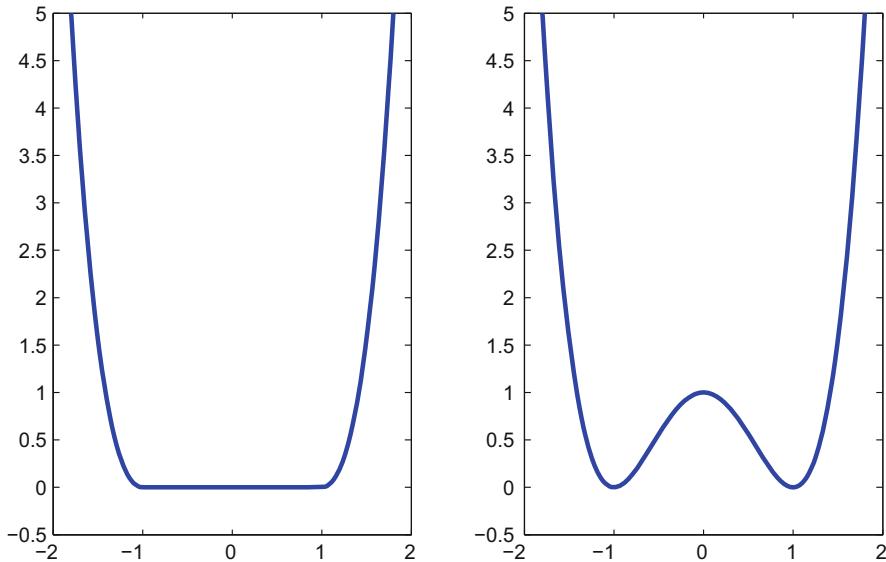


Fig. 29.4 Compositions of convex functions

Proposition 29.10 (Continuity of a Convex Function on an Open Set) *Let $D \subseteq R^n$ be an open convex set and let $f : D \rightarrow R$ be a convex function on D . Then, f is continuous on D .* \square

It must be noted that a convex function over a set which is not open, can be not continuous. As an example, we can consider the function $f : (0, 1] \rightarrow R$ such that $f(x) = 0$ for $x \in (0, 1)$ and $f(1) = 1$, which is convex on $(0, 1]$ but not continuous.

29.4 Convexity of Differentiable Functions

If f is a differentiable function we can give necessary and sufficient convexity conditions through the use of the derivatives. We will confine ourselves to the case of continuously differentiable functions and we refer to the literature for more general conditions.

A first condition, based on the use of the first order derivatives is given in the next proposition.

Proposition 29.11 (Necessary and Sufficient Convexity Conditions 1) *Let $C \subseteq \mathbb{R}^n$ be a convex set and let f be a continuously differentiable real function on an open set containing C . Then f is convex on C if and only if, for all pairs $x, y \in C$ we have:*

$$f(y) \geq f(x) + \nabla f(x)^T(y - x). \quad (29.4)$$

Moreover, f is strictly convex on C if and only if, for all pairs $x, y \in C$, with $y \neq x$, we have:

$$f(y) > f(x) + \nabla f(x)^T(y - x). \quad (29.5)$$

Proof We show first *necessity*. Assume that f is *convex* on the set C and let $x, y \in C$ and

$$0 < \lambda \leq 1.$$

Then we can write:

$$f(x + \lambda(y - x)) = f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y).$$

As $\lambda > 0$, we obtain

$$\left(\frac{f(x + \lambda(y - x)) - f(x)}{\lambda} \right) \leq f(y) - f(x).$$

Recalling the definition of directional derivative, taking limits for $\lambda \rightarrow 0^+$, we have

$$\nabla f(x)^T(y - x) \leq f(y) - f(x),$$

that implies

$$f(y) \geq f(x) + \nabla f(x)^T(y - x),$$

which establish necessity.

Suppose now that f is *strictly convex* on C . If $x, y \in C$ with $x \neq y$ and $0 < \lambda < 1$ we can write:

$$\left(\frac{f(x + \lambda(y - x)) - f(x)}{\lambda} \right) < f(y) - f(x). \quad (29.6)$$

As f is convex, (strict convexity obviously implies convexity), by the results established above, condition (29.4) must hold for the pair (x, z) with $z = x + \lambda(y - x)$, so that we obtain

$$f(x + \lambda(y - x)) - f(x) \geq \lambda \nabla f(x)^T (y - x). \quad (29.7)$$

Then, by (29.6) and (29.7) we have:

$$f(y) - f(x) > \left(\frac{f(x + \lambda(y - x)) - f(x)}{\lambda} \right) \geq \nabla f(x)^T (y - x),$$

which proves condition (29.5).

To establish *sufficiency*, let us assume that (29.4) holds for every pair $x, y \in C$ and let $z = (1 - \lambda)x + \lambda y$ with $0 < \lambda < 1$. By convexity of C , we have $z \in C$. From condition (29.4), referred to the pairs (z, x) and (z, y) , it follows that:

$$f(x) \geq f(z) + \nabla f(z)^T (x - z),$$

$$f(y) \geq f(z) + \nabla f(z)^T (y - z).$$

Then, multiplying the first inequality by $1 - \lambda$ and the second by λ , and summing the two inequalities, we obtain:

$$\begin{aligned} (1 - \lambda)f(x) + \lambda f(y) &\geq (1 - \lambda)f(z) + \lambda f(z) + ((1 - \lambda)(x - z) + \lambda(y - z))^T \nabla f(z) \\ &= f(z) + ((1 - \lambda)(x - z) + \lambda(y - z))^T \nabla f(z) \\ &= f(z) + ((1 - \lambda)x + \lambda y - z)^T \nabla f(z) = f(z) \end{aligned}$$

and this shows that f is convex. Using similar reasonings, using (29.5) and introducing strict inequalities when required, we can establish sufficiency of (29.5) for strict convexity. \square

From a geometric point of view, the condition given in the preceding proposition implies that a function is convex on C if and only if, at every point $y \in C$, the value $f(y)$ is not inferior to the ordinates of the points in a plane tangent to the graph of f at any point x of C . This is shown in Fig. 29.5.

In the next proposition we give convexity conditions based on second order derivatives.

Proposition 29.12 (Necessary and Sufficient Convexity Conditions 2) *Let $C \subseteq R^n$ be an open convex set, and assume that f is a real function twice continuously differentiable on C . Then f is convex on C if and only if, for every $x \in C$, the matrix $\nabla^2 f(x)$ is positive semidefinite.*

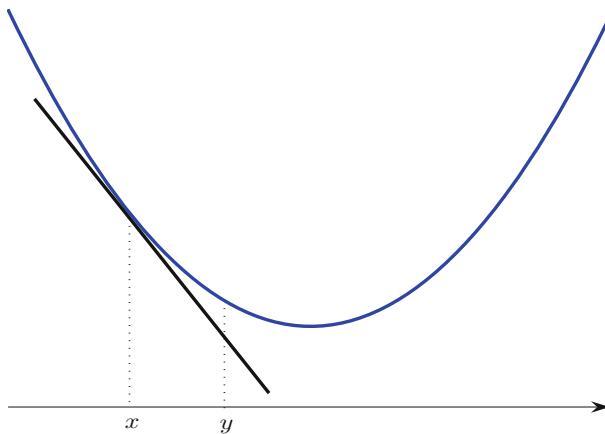


Fig. 29.5 Convexity condition

Proof Consider first necessity. Let $x \in C$ and let $y \neq 0$ an arbitrary vector in R^n . As C is open, we can find $\bar{\lambda} > 0$ sufficiently small to have $x + \lambda y \in C$ for all $0 < \lambda < \bar{\lambda}$. By Proposition 29.11, given the pair $(x + \lambda y, x)$ we have:

$$f(x + \lambda y) - f(x) - \lambda \nabla f(x)^T y \geq 0.$$

On the other hand, as f is twice continuously differentiable, we can write:

$$f(x + \lambda y) = f(x) + \lambda y^T \nabla f(x) + \frac{1}{2} \lambda^2 y^T \nabla^2 f(x) y + \beta(x, \lambda y)$$

where

$$\lim_{\lambda \rightarrow 0} \frac{\beta(x, \lambda y)}{\lambda^2} = 0.$$

Then we have

$$\frac{1}{2} \lambda^2 y^T \nabla^2 f(x) y + \beta(x, \lambda y) \geq 0$$

and hence, dividing by λ^2 and taking limits for $\lambda \rightarrow 0$, we obtain:

$$y^T \nabla^2 f(x) y \geq 0,$$

which proves necessity.

Conversely, assume that the matrix $\nabla^2 f$ is positive semidefinite on C and let $x, y \in C$. From Taylor's formula we get:

$$f(y) = f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(w)(y - x),$$

where $w = x + \xi(y - x)$ with $\xi \in (0, 1)$. As $w \in C$ and $\nabla^2 f(w)$ is positive semidefinite, we obtain

$$f(y) \geq f(x) + \nabla f(x)^T(y - x),$$

which show, by Proposition 29.11, that f is convex on C . \square

The assumption that the Hessian is positive definite is a sufficient, but not necessary, condition for strict convexity. Consider, for instance the strictly convex function $y = x^4$ at $x = 0$. Sufficiency is proved in the following proposition.

Proposition 29.13 (Sufficient Condition for Strict Convexity) *Let C be an open convex set, let $f : C \rightarrow \mathbb{R}$ and assume that the Hessian matrix $\nabla^2 f$ is continuous and positive definite on C . Then f is strictly convex on C .* \square

Proof Suppose that $\nabla^2 f$ is positive definite on C and let x, y with $x \neq y$ two points in C . From Taylor's formula, we obtain

$$f(y) = f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(w)(y - x)$$

where $w = x + \xi(y - x)$ with $\xi \in (0, 1)$. As $w \in C$ and $\nabla^2 f(w)$ is positive definite on C , we have

$$(y - x)^T \nabla^2 f(w)(y - x) > 0,$$

so that

$$f(y) > f(x) + \nabla f(x)^T(y - x),$$

which shows, by Proposition 29.11, that f is strictly convex on C . \square

As a consequence of the preceding results, we can establish the following proposition on convexity of a quadratic function.

Proposition 29.14 (Convexity of a Quadratic Function) Let Q an $n \times n$ symmetric matrix and let $f : R^n \rightarrow R$ be the quadratic function

$$f(x) = \frac{1}{2}x^T Qx + c^T x.$$

Then f is convex on R^n if and only if Q is positive semidefinite. □

The assertion follows immediately from the preceding proposition, as Q is the Hessian matrix of f .

In the case of quadratic functions we can give a complete characterization of strict convexity.

Proposition 29.15 (Strict Convexity of a Quadratic Function) Let Q be a symmetric $n \times n$ matrix and let $f : R^n \rightarrow R$ defined by:

$$f(x) = \frac{1}{2}x^T Qx + c^T x.$$

Then f is strictly convex on R^n if and only if Q is positive definite.

Proof Sufficiency follows from the preceding proposition and hence we must only show necessity. Reasoning by contradiction, assume that f is strictly convex but Q is positive semidefinite (this follows from Proposition 29.14) but not positive definite. This implies that there must exist a zero eigenvalue of Q and hence an eigenvector $x \neq 0$ such that $Qx = 0$. If we consider the points x , $y = -x$ and $z = \frac{1}{2}x + \frac{1}{2}y = 0$ we have:

$$0 = f(z) = \frac{1}{2}f(x) + \frac{1}{2}f(y),$$

which would contradict the strict convexity assumption f . □

29.5 Monotonicity Conditions on ∇f

The convexity conditions can also be expressed through *monotonicity conditions* on ∇f . We introduce the following definition.

Definition 29.13 (Monotone Mapping) Let $D \subseteq R^n$ and $F : D \rightarrow R^n$. We say that F is *monotone* on D if we have

$$(F(y) - F(x))^T (y - x) \geq 0,$$

for every pair $x, y \in D$.

F is *strictly monotone* on D if we have

$$(F(y) - F(x))^T (y - x) > 0,$$

for every pair $x, y \in D$ with $x \neq y$. □

In Fig. 29.6 we illustrate the monotonicity of ∇f for a convex function defined on R .

In the next proposition we show that monotonicity of ∇f is actually a necessary and sufficient condition for convexity.

Proposition 29.16 (Convexity Conditions) Let C be an open convex set, let $f : C \rightarrow R$ and assume that ∇f is continuous on C . Then f is convex on C if and only if ∇f is monotone on C , that is, if and only if, for every pair $x, y \in C$ we have

(continued)

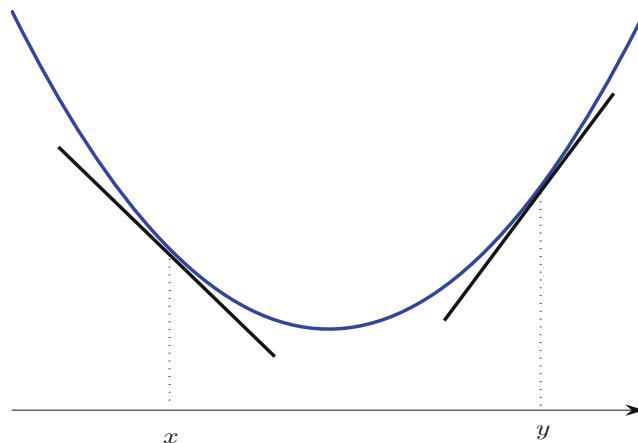


Fig. 29.6 Monotonicity of ∇f

Proposition 29.16 (continued)

$$(\nabla f(y) - \nabla f(x))^T(y - x) \geq 0. \quad (29.8)$$

Moreover, f is strictly convex on C if and only if ∇f is strictly monotone on C , that is, if and only if, for every pair $x, y \in C$ with $x \neq y$ we have

$$(\nabla f(y) - \nabla f(x))^T(y - x) > 0. \quad (29.9)$$

Proof First suppose that f is convex on C and assume that $x, y \in C$. By Proposition 29.11 we have

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

and also

$$f(x) \geq f(y) + \nabla f(y)^T(x - y).$$

Summing both members of the two inequalities above, we get:

$$0 \geq \nabla f(x)^T(y - x) + \nabla f(y)^T(x - y),$$

which implies (29.8).

Now assume that (29.8) is true and let x, y two points of C with $y \neq x$. By the theorem of the mean we can write

$$f(y) = f(x) + \nabla f(x + \lambda(y - x))^T(y - x), \quad (29.10)$$

where $\lambda \in (0, 1)$. On the other hand, by (29.8), (referred to the pair of points $x + \lambda(y - x)$ and x , that belong to C) we have:

$$\nabla f(x + \lambda(y - x))^T\lambda(y - x) \geq \nabla f(x)^T\lambda(y - x),$$

and hence, dividing both members by λ , and taking into account (29.10), we obtain

$$f(y) \geq f(x) + \nabla f(x)^T(y - x),$$

and this, by Proposition 29.11, implies that f is convex.

By repeating a similar reasoning, using strict inequalities when required, we obtain a necessary and sufficient condition of strict convexity. \square

We introduce the following definition.

Definition 29.14 (Strong Convexity) Let $f : R^n \rightarrow R$ be continuously differentiable on R^n . We say that f is *strongly convex* on R^n if ∇f is *strongly monotone* on R^n , that is there exists $\alpha > 0$ such that

$$(\nabla f(y) - \nabla f(x))^T(y - x) \geq \alpha \|y - x\|^2,$$

for every pair $x, y \in R^n$. □

It can be shown (see, e.g. [16], Proposition B.5) that if f is strongly convex then it is also strictly convex and, moreover, that if f is twice continuously differentiable then f is strongly convex if and only if the matrix $\nabla^2 f(x) - \alpha I$ is positive semidefinite for every $x \in R^n$.

29.6 Basic Notions of Generalized Convexity

Various extensions of the notion of convexity has been introduced, by relaxing the conditions that characterize convex functions, but retaining some important property of these functions.

In the sequel we will confine ourselves to state some basic concepts of generalized convexity recalled in the book. We refer the interested reader to [179] for a comprehensive introduction to this subject.

We state first the notions of quasi-convex and quasi-concave functions.

Definition 29.15 (Quasi-Convex and Quasi-Concave Functions) Let $S \subseteq R^n$ be a nonempty convex set and let $f : S \rightarrow R$. The function f is said to be *quasi-convex* on S if, for each pair $x, y \in S$, we have

$$f((1 - \lambda)x + \lambda y) \leq \max\{f(x), f(y)\},$$

for all λ such that $0 \leq \lambda \leq 1$.

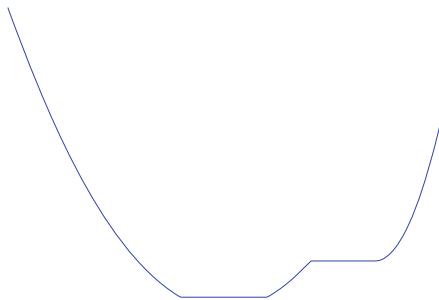
The function $f : S \rightarrow R$ is said to be *quasi-concave* on S if $-f$ is quasi-convex on S , that is if, for each pair $x, y \in S$, we have

$$f((1 - \lambda)x + \lambda y) \geq \min\{f(x), f(y)\},$$

for all λ such that $0 \leq \lambda \leq 1$. □

An example of a quasi-convex functions defined on R is shown in Fig. 29.7.

Fig. 29.7 Quasi-convex function: there exist local non global minimizers



The quasi-convexity of a given function can be characterized through convexity conditions on the level sets of f . In fact, we can state the following proposition.

Proposition 29.17 (Convexity of Level Sets) *Let S be a convex set and let $f : S \rightarrow \mathbb{R}$. Then f is quasi-convex on S if and only if, for all $\alpha \in \mathbb{R}$, the level sets*

$$\mathcal{L}(\alpha) = \{x \in S : f(x) \leq \alpha\}$$

are convex. Similarly, f is quasi-concave on S if and only if, for all $\alpha \in \mathbb{R}$, the upper level sets

$$\mathcal{Q}(\alpha) = \{x \in S : f(x) \geq \alpha\}$$

are convex.

Proof Assume first that f is quasi-convex and let $\alpha \in \mathbb{R}$. If $\mathcal{L}(\alpha) = \emptyset$ the assertion holds vacuously. Thus assume that $\mathcal{L}(\alpha)$ is nonempty and that $x, y \in \mathcal{L}(\alpha)$. This implies, in particular, that

$$\max\{f(x), f(y)\} \leq \alpha.$$

Consider the point If $z = (1 - \lambda)x + \lambda y$ with $\lambda \in [0, 1]$. By convexity of S we have $z \in S$ and, by quasi-convexity of f we have

$$f(z) \leq \max\{f(x), f(y)\}.$$

It follows that $f(z) \leq \alpha$ and this implies the convexity of $\mathcal{L}(\alpha)$.

Suppose now that the level sets $\mathcal{L}(\alpha)$ are convex for all α and let $x, y \in S$. Letting $\alpha = \max\{f(x), f(y)\}$ we have obviously that $x, y \in \mathcal{L}(\alpha)$.

By convexity of $\mathcal{L}(\alpha)$ we have that

$$z = (1 - \lambda)x + \lambda y \in \mathcal{L}(\alpha)$$

for $\lambda \in [0, 1]$, so that $f(z) \leq \alpha = \max\{f(x), f(y)\}$, which proves the quasi-convexity of f .

Quasi-concavity conditions can be obtained by replacing f with $-f$ and the convexity $\mathcal{L}(\alpha)$ with the convexity of the upper level sets $\Omega(\alpha)$. \square

The quasi-convexity of f does not exclude the presence of local non global minimizers, as shown in Fig. 29.7. To exclude this possibility we must refer to a stronger notion of quasi-convexity, often known as *strict quasi-convexity*.

Definition 29.16 (Strict Quasi-Convexity and Quasi-Concavity) Let $S \subseteq R^n$ be a convex set and let $f : S \rightarrow R$. The function f is said to be *strictly quasi-convex* on S if, for each pair $x, y \in S$, such that $f(x) \neq f(y)$ we have

$$f((1 - \lambda)x + \lambda y) < \max\{f(x), f(y)\},$$

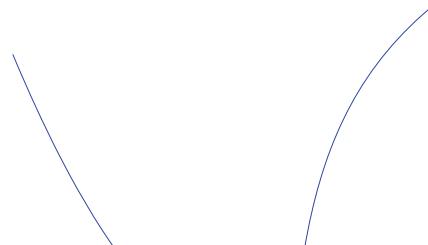
for all $\lambda \in (0, 1)$. Similarly, the function $f : S \rightarrow R$ is said to be *strictly quasi-concave* on S if $-f$ is strictly quasi-convex on S . \square

An example of strictly quasi-convex function is given in Fig. 2.4.

The condition of strict quasi-convexity given in the preceding definition does not guarantee the uniqueness of the optimal solution. In order to guarantee the uniqueness of the global minimizer we must impose a stronger condition, which some authors call *strong quasi-convexity* (Fig. 29.8).

Note however that in many works, especially in the economic literature, other authors call *strict quasi-convexity* the condition defined here as *strong quasi-convexity*.

Fig. 29.8 Strictly quasi convex function: every local minimizer is a global minimizer



Definition 29.17 (Strong Quasi-Convexity and Strong Quasi-Concavity)

Let $S \subseteq R^n$ be a convex set and let $f : S \rightarrow R$. The function f is said to be *strongly quasi-convex* on S if, for all $x, y \in S$ such that $x \neq y$, we have

$$f((1 - \lambda)x + \lambda y) < \max\{f(x), f(y)\},$$

for all $\lambda \in (0, 1)$.

The function $f : S \rightarrow R$ is said to be *strongly quasi-concave* on S if $-f$ is strongly quasi-convex on S , or, equivalently, if, for all $x, y \in S$ such that $x \neq y$, we have

$$f((1 - \lambda)x + \lambda y) > \min\{f(x), f(y)\},$$

for all λ such that $0 < \lambda < 1$. □

The definition above is illustrated in Fig. 29.9.

In order to establish a connection of the notion of strong quasi-convexity with the structure of the level sets we first introduce the following definition.

Definition 29.18 (Strictly Convex Set) A convex set $S \subseteq R^n$ is said to be *strictly convex* if, for all pairs x, y on the boundary of S , with $x \neq y$, all the points $z = (1 - \lambda)x + \lambda y$ with $0 < \lambda < 1$ are interior points of S . □

It can be easily verified that, if f is strongly quasi-convex then the level sets are strictly convex.

When f is differentiable, we can observe that the preceding conditions of generalized convexity do not guarantee that stationary points are global minimizers.

We can introduce a different generalization of convexity that yields this implication. In particular, we can introduce the following definitions.

Fig. 29.9 Strongly quasi-convex function: the minimum point is unique

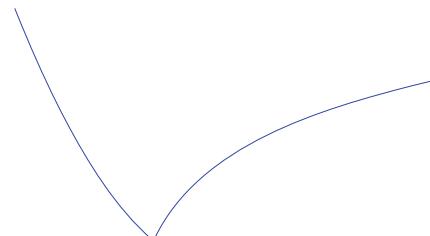
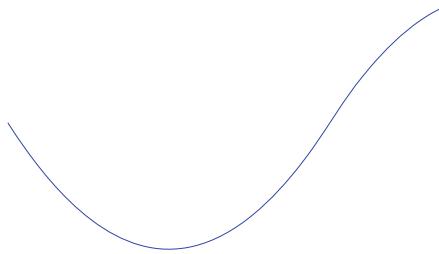


Fig. 29.10 Strictly pseudo convex function



Definition 29.19 ((Strictly) Pseudoconvex and Pseudoconcave Functions)

Let $D \subseteq R^n$ be an open set, let S be a convex set contained in D , let $f : D \rightarrow R$ and assume that ∇f is continuous on D . Then f is said to be *pseudoconvex* on S if, for all pairs $x, y \in S$, we have that $\nabla f(x)^T(y - x) \geq 0$ implies $f(y) \geq f(x)$.

f is strictly pseudoconvex on S if, for all pairs $x, y \in S$ with $x \neq y$ we have that $\nabla f(x)^T(y - x) \geq 0$ implies $f(y) > f(x)$.

The function f is said to be (strictly) pseudoconcave if $-f$ is (strictly) pseudoconvex. \square

A strictly pseudoconvex function is shown in Fig. 29.10. It can be easily verified that when f is a differentiable pseudoconvex function defined on S , any critical point of f in S is a global minimizer. See, for instance, Chap. 4. In particular, in the unconstrained case every stationary point of a pseudoconvex function on R^n is a global minimizer of f , which is also the unique global minimizer, if f is strictly pseudoconvex.

We state here, following [12, 179], some connections between the different definitions of convexity and generalized convexity introduced in this chapter. In particular, we refer to a function f defined on some open convex set $S \subseteq R^n$ and we assume that f is continuously differentiable when a pseudoconvex function is considered.

- f strictly convex $\Rightarrow f$ convex;
- f convex $\Rightarrow f$ pseudoconvex;
- f convex $\Rightarrow f$ quasi-convex;
- f pseudoconvex $\Rightarrow f$ strictly quasi-convex;
- f strictly quasi-convex $\Rightarrow f$ quasi-convex (f lower semi-continuous);
- f strongly quasi-convex $\Rightarrow f$ quasi-convex;

(continued)

- f strictly convex $\Rightarrow f$ strictly pseudoconvex;
- f strictly pseudoconvex $\Rightarrow f$ pseudoconvex;
- f strictly pseudoconvex $\Rightarrow f$ strongly quasi-convex;
- f strongly quasi-convex $\Rightarrow f$ strictly quasi-convex.

We remark that a logical implication between two propositions A and B of the form $A \Rightarrow B$ is equivalent to $\bar{B} \Rightarrow \bar{A}$, where \bar{B} , \bar{A} indicate the negation of B , A respectively. Thus the implication:

$$f \text{ strictly pseudoconvex} \Rightarrow f \text{ strongly quasi-convex}.$$

is equivalent to

$$f \text{ is not strongly quasi-convex} \Rightarrow f \text{ is not strictly pseudoconvex}.$$

This implication is proved below.

Proposition 29.18 *Let $D \subseteq R^n$ be an open set, let $S \subseteq D$ be a convex set, let $f : D \rightarrow R$ and assume that ∇f is continuous on D and that f is strictly pseudoconvex on S . Then f is strongly quasi-convex on S .*

Proof Reasoning by contradiction, let us assume that f is not strongly quasi-convex on S . Then, there must exist $x, y \in S$, with $x \neq y$, and $\lambda \in (0, 1)$ such that $f(z) \geq \max\{f(x), f(y)\}$ with $z = (1 - \lambda)x + \lambda y$. As $f(x) \leq f(z)$, the strict pseudoconvexity of f implies that $\nabla f(z)^T(x - z) < 0$, which implies, in turn, that $\nabla f(z)^T(x - y) < 0$. Similarly, as $f(y) \leq f(z)$, using the same reasoning we get $\nabla f(z)^T(y - x) < 0$ so that we obtain a contradiction between the two inequalities obtained. \square

The reader is invited to prove the other implications displayed above.

References

1. Al-Baali, M.: Descent property and global convergence of the Fletcher-Reeves method with inexact line searches. *IMA J. Numer. Anal.* **5**, 121–124 (1985)
2. Al-Baali, M., Grandinetti, L., Pisacane, O.: Damped techniques for the limited memory BFGS method for large-scale optimization. *J. Optim. Theory Appl.* **161**(2), 688–609 (2014)
3. Alexeev, V., Tikhomirov, V., Fomine, S.: *Commande Optimale*. MIR, Moscou (1982)
4. Amaldi, E., Kann, V.: On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theor. Comput. Sci.* **209**(1–2), 237–260 (1998)
5. Armijo, L.: Minimization of functions having continuous partial derivatives. *Pac. J. Math.* **16**, 1–3 (1966)
6. De Asmundis, R., Di Serafino, D., Hager, W.W., Toraldo, G., Zhang, H.: An efficient gradient method using the Yuan steplength. *Comput. Optim. Appl.* **59**(3), 541–563 (2014)
7. Audet, C., Hare, W.: *Derivative-Free and Blackbox Optimization*. Springer, Cham (2017)
8. Auslender, A.: Asymptotic properties of the fenchel dual functional and applications to decomposition problems. *J. Optim. Theory Appl.* **73**, 427–449 (1992)
9. Bach, F.: Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *J. Mach. Learn. Res.* **15**, 595–627 (2014)
10. Balinski, M.L., Tucker, A.W.: Duality theory of linear programs: a constructive approach with applications. *SIAM Rev.* **11**(3), 347–377 (1969)
11. Barzilai, J., Borwein, M.J.: Two point step size gradient method. *IMA J. Numer. Anal.* **8**, 141–188 (1988)
12. Bazaraa, M., Sherali, H., Shetty, C.: *Nonlinear Programming, Theory, and Applications*, 2nd edn. Wiley, New York (1993)
13. Bellavia, S., Gasparo, M.G., Macconi, M.: A switching-method for nonlinear system. *J. Comput. Appl. Math.* **7**, 83–93 (1996)
14. Ben-Tal, A., Nemirovski, A.: *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, Philadelphia (2001)
15. Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York (1982)
16. Bertsekas, D.P.: *Nonlinear Programming*, 2nd edn. Athena Scientific, Belmont (1999)
17. Bertsekas, D.P.: *Convex Optimization Theory*. Athena Scientific, Belmont (2009)
18. Bertsekas, D.P., Tsitsiklis, J.N.: *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs (1989)
19. Bertsimas, D., Tsitsiklis, J.: *Introduction to Linear Optimization*. Athena Scientific, Belmont (1997)

20. Best, M.J., Bräuninger, J., Ritter, K., Robinson, S.M.: A globally and quadratically convergent algorithm for general nonlinear programming problems. *Computing* **26**(2), 141–153 (1981)
21. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **10**, 1196–1211 (2000)
22. Birgin, E.G., Gardenghi, J.L., Martínez, J.M., Santos, S.A., Toint, P.L.: Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Math. Program. Ser. A* **163**, 359–368 (2017)
23. Birgin, E.G., Martinez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM, Philadelphia (2014)
24. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
25. Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
26. Bogle, I.D.L., Perkins, J.D.: A new sparsity preserving quasi-Newton update for solving nonlinear equations. *SIAM J. Sci. Stat. Comput.* **11**(4), 621–630 (1990)
27. Bomze, I.M., Rinaldi, F., Zeffiro, D.: Frank - Wolfe and friends: a journey into projection-free first-order optimization methods. *4OR* **19**, 313–345 (2021)
28. Bonettini, S.: A nonmonotone inexact Newton method. *Optim. Methods Softw.* **20**(4–5), 475–491 (2005)
29. Bonettini, S.: Inexact block coordinate descent methods with application to the nonnegative matrix factorization. *IMA J. Numer. Anal.* **31**, 1431–1452 (2011)
30. Bonnans, J.F., Panier, E.R., Tits, A.L., Zhou, J.L.: Avoiding the Maratos effect by means of a nonmonotone line search. II. inequality constrained problems—feasible iterates. *SIAM J. Numer. Anal.* **29**(4), 1187–1202 (1992)
31. Bonnans, J.F., Gilbert, J.C., Lemarechal, C., Sagastizabal, C.A.: Numerical Optimization: Theoretical and Practical Aspects. Springer, Berlin (2006)
32. Bottou, L., Bengio, Y.: Convergence properties of the k-means algorithms. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in Neural Information Processing Systems, vol. 7, pp. 585–592. MIT Press, Cambridge (1994)
33. Braess, D.: Nonlinear Approximation Theory. Springer, Berlin (1986)
34. Brown, P.N.: A local convergence theory for combined inexact-Newton/finite difference methods. *SIAM J. Numer. Anal.* **24**, 407–434 (1987)
35. Brown, P.N., Saad, Y.: Convergence theory of nonlinear Newton-Krylov algorithms. *SIAM J. Optim.* **4**, 297–330 (1994)
36. Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. *Math. Comput.* **19**, 577–593 (1965)
37. Buzzi, C., Grippo, L., Sciandrone, M.: Convergent decomposition techniques for training RBF neural networks. *Neural Comput.* **13**(8), 1891–1920 (2001)
38. Byrd, R.H., Nocedal, J.: A tool for the analysis of quasi-newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.* **26**, 727–739 (1989)
39. Cartis, C., Gould, N.I.M., Toint, P.L.: On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM J. Optim.* **20**(6), 2833–2852 (2010)
40. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic overestimation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program. Ser. A* **127**, 245–295 (2011)
41. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic overestimation methods for unconstrained optimization. Part II: worst-case function-evaluation complexity. *Math. Program. Ser. A* **130**, 295–319 (2011)
42. Cartis, C., Gould, N.I.M., Toint, P.L.: Evaluation Complexity of Algorithms for Nonconvex Optimization Theory, Computation, and Perspectives. SIAM, Philadphelia (2022)
43. Cartis, C., Sanpalo, P.R., Toint, P.L.: Worst-case evaluation complexity of non-monotone gradient-related algorithms for unconstrained optimization. *Optimization* **64**(5), 1349–1361 (2015)
44. Cassioli, A., Di Lorenzo, D., Sciandrone, M.: On the convergence of inexact block coordinate descent methods for constrained optimization. *Eur. J. Oper. Res.* **231**, 274–281 (2013)

45. Céa, J.: Optimisation. Dunod, Paris (1971)
46. Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., Pedersen, H.C.: The watchdog technique for forcing convergence in algorithms for constrained optimization. *Math. Program. Study* **16**, 1–17 (1982)
47. Chen, P.H., Fan, R.E., Lin, C.J.: Working set selection using second order information for training support vector machines. *J. Mach. Learn. Res.* **6**, 1889–1918 (2005)
48. Choi, T.D., Kelley, C.T.: Superlinear convergence and implicit filtering. *SIAM J. Optim.* **10**, 1149–1162 (2000)
49. Cocchi, G., Galligari, A., Picca Nicolino, F., Piccialli, V., Schoen, F., Sciandrone, M.: Scheduling the Italian national volleyball tournament. *Interfaces* **48**, 271–284 (2018)
50. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. MPS/SIAM Series on Optimization. SIAM, Philadelphia (2000)
51. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. SIAM, Philadelphia (2009)
52. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
53. Custodio, A.L., Dennis, J.E., Jr., Vicente, L.N.: Using simplex gradients of nonsmooth functions in direct search methods. *IMA J. Numer. Anal.* **20**, 770–784 (2008)
54. Tamiz, M., Jones, D., et al.: Practical Goal Programming, vol. 141. Springer, Berlin (2010)
55. Dai, Y.H., Liao, L.Z.: R-linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numer. Anal.* **2**, 1–10 (2002)
56. Dai, Y.H., Fletcher, R.: Projected Barzilai-Borwein methods for large scale box-constrained quadratic programming. *Numer. Math.* **100**, 21–47 (2005)
57. Daniel, J.W.: The conjugate gradient method for linear and nonlinear operator equations. *SIAM J. Numer. Anal.* **4**(1), 10–26 (1967)
58. Dau, Y.H.: Convergence properties of the BFGS algorithm. *SIAM J. Optim.* **13**:693–701, 2002.
59. Davidon, W.C.: Variable metric method for minimization. Technical Report, Argonne National Lab, IL, USA (1966)
60. Davidon, W.C.: Variable metric method for minimization. *SIAM J. Optim.* **1**(1), 1–17 (1991)
61. De Leone, R., Gaudioso, M., Grippo, L.: Stopping criteria for linesearch methods without derivatives. *Math. Program.* **30**, 285–300 (1984)
62. De Simone, V., Di Serafino, D., Gondzio, J., Pougkakiotis, S., Viola, M.: Sparse approximations with interior point methods. Technical Report, arXiv (2021)
63. Defazio, A., Bach, F., Lacoste-Julien, S.: Saga: a fast incremental gradient method with support for non-strongly convex composite objectives. In: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS’14), vol. 1, pp. 1646–1654. MIT Press, Cambridge (2014)
64. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. *SIAM J. Numer. Anal.* **19**, 400–408 (1982)
65. Deng, N.Y., Xiao, Y., Zhou, F.J.: Nonmonotonic trust region algorithm. *J. Optim. Theory Appl.* **76**(2), 259–285 (1993)
66. Dennis, J.E., Moré, J.J.: Quasi-Newton methods, motivation and theory. *SIAM Rev.* **19**, 46–89 (1977)
67. Dennis, J.E., Schnabel, R.B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Prentice-Hall, Englewood Cliffs (1983)
68. Di Pillo, G.: Exact penalty methods. In: Spedicato, E. (ed.) Algorithms for Continuous Optimization, vol. 434, pp. 209–253. Springer, Dordrecht (1994)
69. Di Pillo, G., Facchinei, F., Grippo, L.: An RQP algorithm using a differentiable exact penalty function for inequality constrained problems. *Math. Program.* **55**, 49–68 (1992)
70. Di Pillo, G., Grippo, L.: A new class of augmented Lagrangians in nonlinear programming. *SIAM J. Control Optim.* **17**(5), 618–628 (1979)
71. Di Pillo, G., Grippo, L.: A new augmented Lagrangian function for inequality constraints in nonlinear programming problems. *J. Optim. Theory Appl.* **36**(4), 495–519 (1982)

72. Di Pillo, G., Grippo, L.: A continuously differentiable exact penalty function algorithm for nonlinear programming problems with inequality constraints. *SIAM J. Control Optim.* **23**, 72–84 (1985)
73. Di Pillo, G., Grippo, L.: An exact penalty method with global convergence properties for nonlinear programming problems. *Math. Program.* **36**, 1–18 (1986)
74. Di Pillo, G., Grippo, L.: On the exactness of a class of nondifferentiable penalty functions. *J. Optim. Theory Appl.* **53**(3), 399–410 (1988)
75. Di Pillo, G., Grippo, L.: Exact penalty functions in constrained optimization. *SIAM J. Control Optim.* **27**, 1333–1360 (1989)
76. Di Pillo, G., Grippo, L., Lucidi, S.: A smooth method for the finite minimax problem. *Math. Program.* **60**, 187–214 (1993)
77. Di Pillo, G., Liuzzi, G., Lucidi, S., Palagi, L.: A truncated Newton method in an augmented Lagrangian framework for nonlinear programming. *Comput. Optim. Appl.* **45**(2), 311–352 (2010)
78. Di Pillo, G., Lucidi, S.: An augmented Lagrangian function with improved exactness properties. *SIAM J. Optim.* **12**(2), 376–406 (2002)
79. Dikin, I.I.: Iterative solution of problems of linear and quadratic programming. *Sov. Math. Dokl.* **8**, 674–675 (1967)
80. Dirkse, S.P., Ferris, M.C.: The path solver: a nonmonotone stabilization scheme for mixed complementarity problems. *Optim. Methods Softw.* **5**(2), 123–156 (1995)
81. Dixon, L.C.W.: Variable metric algorithms: necessary and sufficient conditions for identical behavior on nonquadratic functions. *J. Optim. Theory Appl.* **10**, 34–40 (1972)
82. Martínez, J.M., Birgin, E.G., Raydan, M.: Spectral projected gradient methods: review and perspectives. *J. Stat. Softw.* **60**, 1–21 (2014)
83. Eisenstat, S.C., Walker, H.F.: Globally convergent inexact Newton methods. *SIAM J. Optim.* **4**, 16–32 (1994)
84. Evtushenko, Y.G.: Numerical Optimization Techniques. Optimization Software, Inc. Publication Division, New York (1985)
85. Facchinei, F., Lucidi, S.: A class of penalty functions for optimization problems with bound constraints. *Optimization* **26**, 239–259 (1992)
86. Facchinei, F., Lucidi, S.: Quadratically and superlinearly convergent algorithms for the solution of inequality constrained minimization problems. *J. Optim. Theory Appl.* **85**(2), 265–289 (1995)
87. Facchinei, F., Pang, J.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, New York (2003)
88. Fang, H., O’Leary, D.: Modified Cholesky algorithms: a catalog with new approaches. *Math. Program.* **115**, 319–349 (2008)
89. Fasano, G., Lucidi, S.: A nonmonotone truncated Newton-Krylov method employing negative curvature directions for large scale unconstrained optimization. *Optim. Lett.* **3**(4), 521–535 (2009)
90. Ferris, M.C., Lucidi, S.: Nonmonotone stabilization methods for nonlinear equations. *J. Optim. Theory Appl.* **81**, 815–832 (1994)
91. Ferris, M.C., Mangasarian, O.L., Wright, S.J.: Linear Programming with MATLAB. SIAM, Philadelphia (2007)
92. Fiacco, A.V., McCormick, G.P.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley, New York (1968)
93. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (1987)
94. Fletcher, R.: On the Barzilai-Borwein method. In: Optimization and Control with Applications, pp. 235–256. Springer, Boston (2005)
95. Fletcher, R., Powell, M.J.D.: A rapidly convergent descent method for minimization. *Comput. J.* **6**(2), 163–168 (1963)
96. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. *Comput. J.* **6**, 163–168 (1964)

97. Fletcher, R.E.: A class of methods for nonlinear programming with termination and convergence properties. In: Abadie, J. (ed.) Integer and Nonlinear Programming, pp. 157–173. North-Holland, Amsterdam (1970)
98. Florian, M., Hearn, D.W.: Traffic Assignment: Equilibrium Models, pp. 571–592. Springer, New York (2008)
99. Florian, M.S., Hearn, D.: Network Equilibrium Models and Algorithms. In: Ball, M.O., Magnanti, T.L., Momma, C.L., Nemhauser, G.L. (eds.) Handbooks in OR and MS, vol. 8, pp. 485–550. North-Holland, Amsterdam (1995)
100. Frank, M., Wolfe, P.: An algorithm for quadratic programming. *Naval Res. Logist. Quart.* **3**, 95–110 (1956)
101. Friedlander, A., Martinez, J.M., Molina, B., Raydan, M.: Gradient method with retards and generalizations. *SIAM J. Numer. Anal.* **36**, 275–289 (1999)
102. Galli, L., Galligari, A., Sciandrone, M.: A unified convergence framework for nonmonotone inexact decomposition methods. *Comput. Optim. Appl.* **75**, 113–144 (2020)
103. Galvan, G., Lapucci, M., Lin, C.J., Sciandrone, M.: A two-level decomposition framework exploiting first and second order information for svm training problems. *J. Mach. Learn. Res.* **22**, 1–38 (2021)
104. García-Palomares, U.M., García-Urrea, I.J., Rodríguez-Hernández, P.S.: On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization. *Optim. Methods Softw.* **28**(6), 1233–1261 (2013)
105. Garey, M.R., Johnson, D.S.: Computers and Intractability. W.H. Freeman, New York (1979)
106. Gasparo, M.: A nonmonotone hybrid method for nonlinear systems. *Optim. Methods Softw.* **13**, 79–84 (2000)
107. Gasparo, M.G., Papini, A., Pasquali, A.: Nonmonotone algorithms for pattern search methods. *Numer. Algorithms* **28**(1), 171–186 (2001)
108. Gasparo, M.G., Pieraccini, S., Armellini, A.: An infeasible interior-point method with nonmonotonic complementarity gaps. *Optim. Methods Softw.* **17**(4), 561–586 (2002)
109. Güler, O.: Foundations of Optimization. Springer, New York (2010)
110. Giannessi, F., Nicolucci, F.: Connections between nonlinear and integer programming problems. In: Istituto Nazionale di Alta Matematica, Symposia Mathematica, pp. 161–176. Academic Press, Cambridge (1976)
111. Gilbert, J., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.* **2**, 21–42 (1992)
112. Gilbert, J.C.: Newton’s methods in constrained optimization. In: Lemarechal, C., Bonnans, J.-F., Gilbert, J.C., Sagastizabal, C.A. (eds.) Numerical Optimization, pp. 157–281. Springer, Berlin (1997)
113. Gill, P.E., Murray, W.: Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.* **7**, 311–350 (1974)
114. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)
115. Girsanov, I.V.: Lectures on Mathematical Theory of Extremum. Springer, Berlin (1972)
116. Glad, T., Polak, E.: A multiplier method with automatic limitation of penalty growth. *Math. Program.* **17**, 251–269 (1976)
117. Goldstein, A.A.: Chaucy’s method of minimization. *Numer. Math.* **4**, 146–150 (1962)
118. Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and Combinatorial Optimization. Springer, Berlin (1991)
119. Greenbaum, A.: Iterative Methods for Solving Linear Systems. Society for Industrial and Applied Mathematics, Philadelphia (1997)
120. Griewank, A.: The modification of Newton’s method for unconstrained optimization by bounding cubic terms. Technical Report, Department of Applied Mathematics and Theoretical Physics, University of Cambridge (1981)
121. Grippo, L.: Convergent on-line algorithms for supervised learning in neural networks. *IEEE Trans. Neural Netw.* **11**, 1284–1299 (2000)
122. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton’s method. *SIAM J. Numer. Anal.* **23**, 707–716 (1986)

123. Grippo, L., Lampariello, F., Lucidi, S.: Global convergence and stabilization of unconstrained minimization methods without derivatives. *J. Optim. Theory Appl.* **56**, 385–406 (1988)
124. Grippo, L., Lampariello, F., Lucidi, S.: A truncated Newton method with nonmonotone linesearch for unconstrained optimization. *J. Optim. Theory Appl.* **60**, 401–419 (1989)
125. Grippo, L., Lampariello, F., Lucidi, S.: A class of nonmonotone stabilization methods in unconstrained optimization. *Numer. Math.* **59**, 779–805 (1991)
126. Grippo, L., Lucidi, S.: A differentiable exact penalty function for bound constrained quadratic programming problems. *Optimization* **22**(4), 557–578 (1991)
127. Grippo, L., Lucidi, S.: A globally convergent version of the Polak-Ribière conjugate gradient method. *Math. Program.* **78**, 375–391 (1997)
128. Grippo, L., Lucidi, S.: Convergence conditions, line search algorithms and trust region implementations for the Polak-Ribière conjugate gradient method. *Optim. Methods Softw.* **20**, 71–98 (2005)
129. Grippo, L., Manno, A., Sciandrone, M.: Decomposition techniques for multilayer perceptron training. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(11), 2146–2159 (2016)
130. Grippo, L., Palagi, L., Piccialli, V.: An unconstrained minimization method for solving low-rank SDP relaxations of the maxcut problem. *Math. Program.* **126**(1), 119–146 (2011)
131. Grippo, L., Rinaldi, F.: A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations. *Comput. Optim. Appl.* **60**, 1–33 (2015)
132. Grippo, L., Sciandrone, M.: Globally convergent block-coordinate techniques for unconstrained optimization. *Optim. Methods Softw.* **10**, 587–637 (1999)
133. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Oper. Res. Lett.* **26**, 127–136 (2000)
134. Grippo, L., Sciandrone, M.: Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. *Comput. Optim. Appl.* **23**, 143–169 (2002)
135. Grippo, L., Sciandrone, M.: Nonmonotone derivative-free methods for nonlinear equations. *Comput. Optim. Appl.* **27**, 297–328 (2007)
136. Grippo, L., Sciandrone, M.: Nonmonotone globalization of the finite-difference newton-GMRES method for nonlinear equations. *Optim. Methods Softw.* **25**, 971–999 (2010)
137. Hager, W.W., Zhang, H.: A survey of nonlinear conjugate gradient methods. In: Ceragioli, F. et al. (ed.) *System Modeling and Optimization*, pp. 67–82. Springer, Berlin (1996)
138. Han, S.P., Mangasarian, O.L.: Exact penalty functions in nonlinear programming. *Math. Program.* **17**, 251–269 (1979)
139. Hastie, T., Tibshirani, R., Friedman, J.H.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Springer, Berlin (2009)
140. Haykin, S.: *Neural Networks: A Comprehensive Foundation*, 2nd edn. Prentice Hall, Upper Saddle River (1999)
141. Hestenes, M.R.: Multiplier and gradient methods. *J. Optim. Theory Appl.* **4**(5), 303–320 (1969)
142. Hestenes, M.R.: *Conjugate Direction Methods in Optimization*. Spinger, New York (1980)
143. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bureau Standards* **49**, 409–436 (1952)
144. Hooke, R., Jeeves, T.A.: Direct search solution of numerical and statistical problems. *J. Assoc. Comput. Mach.* **8**, 212–221 (1961)
145. Horst, R., Pardalos, P.M., Thoai, N.V.: *Introduction to Global Optimization*, 2nd edn. Kluwer Academic Publishers, Dordrecht (2000)
146. Horst, R., Tuy, H.: *Global Optimization: Deterministic Approaches*. Springer, Berlin (2013)
147. Kalman, R.E.: A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82**, 35–45 (1960)
148. Kantorovich, L.V.: On Newton's method. *Trudy Mat. Inst. Steklow* **28**, 104–144 (1945)
149. Kantorovich, L.V., Akilov, G.P.: *Analisi Funzionale*. MIR-Editori Riuniti, Roma (1980)
150. Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pp. 302–311 (1984)

151. Keerthi, S., Gilbert, E.: Convergence of a generalized SMO algorithm for SVM. *Mach. Learn.* **46**, 351–360 (2002)
152. Kelley, C.T.: *Iterative Methods for Linear and Nonlinear Equations*. SIAM Publications, Philadelphia (1995)
153. Kelley, C.T.: Detection and remediation of stagnation in the Nelder-Mead algorithm using sufficient decrease condition. *SIAM J. Optim.* **10**, 43–55 (1999)
154. Kelley, C.T.: *Iterative Methods for Optimization*. SIAM Publications, Philadelphia (1999)
155. Khachian, L.G.: A polynomial algorithm in linear programming. *Sov. Math. Dokl.* **20**, 191–194 (1979)
156. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations (ICLR’15), San Diego, May 7–9, 2015, Conference Track Proceedings (2015)
157. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspective on some classical and modern methods. *SIAM Rev.* **45**, 385–482 (2003)
158. Koulikov, L.: *Algebre et theorie des nombres*. Editions MIR, Moscou (1982)
159. La Cruz, W., Martinez, J.M., Raydan, M.: Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Math. Comput.* **75**, 1429–1448 (2006)
160. La Cruz, W., Raydan, M.: Nonmonotone spectral methods for large-scale nonlinear systems. *Optim. Methods Softw.* **18**, 583–599 (2003)
161. Lampariello, F., Sciandrone, M.: Global convergence technique for the Newton method with periodic hessian evaluation. *J. Optim. Theory Appl.* **111**, 341–358 (2001)
162. Lemaréchal, C., Hiriart Urruty, J.B.: *Convex Analysis and Minimization Algorithms*. Springer, Berlin (1993)
163. Li, D.-H., Fukushima, M.: A derivative-free line search and global convergence of broyden-like method for nonlinear equations. *Optim. Methods Softw.* **13**(3), 181–201 (2000)
164. Lin, C.J.: On the convergence of the decomposition method for support vector machines. *IEEE Trans. Neural Netw.* **12**, 1288–1298 (2001)
165. Lin, C.J.: Asymptotic convergence of an smo algorithm without any assumptions. *IEEE Trans. Neural Netw.* **13**, 248–250 (2002)
166. Liu, D.C., Nocedal, J.: On the limited-memory BFGS method for large scale optimization. *Math. Program.* **45**, 503–528 (1989)
167. Locatelli, M., Schoen, F.: *Global Optimization: Theory, Algorithms and Applications*. SIAM, Philadelphia (2013)
168. Pontriaguine, L., Boltianski, V., Gamkrelidze, R., Michtchenko, E.: *Theorie Mathematique des Processus Optimaux*. MIR Publishers, Moscow (1974)
169. Lucidi, S.: New results on a class of exact augmented Lagrangians. *J. Optim. Theory Appl.* **58**, 259–282 (1988)
170. Lucidi, S.: New results on a continuously differentiable exact penalty function. *SIAM J. Optim.* **2**, 558–574 (1992)
171. Lucidi, S., Palagi, L., Roma, M.: On some properties of quadratic programs with a convex quadratic constraint. *SIAM J. Optim.* **8**(1), 105–122 (1998)
172. Lucidi, S., Risi, A., Palagi, L., Sciandrone, M.: A convergent hybrid decomposition algorithm model for SVM training. *IEEE Trans. Neural Netw.* **20**, 1055–1060 (2009)
173. Lucidi, S., Rochetich, F., Roma, M.: Curvilinear stabilization techniques for truncated Newton methods in large scale unconstrained optimization. *SIAM J. Optim.* **8**, 916–939 (1998)
174. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. *Comput. Optim. Appl.* **21**, 119–142 (2002)
175. Lucidi, S., Sciandrone, M.: On the global convergence of derivative free methods for unconstrained optimization. *SIAM J. Optim.* **13**, 97–116 (2002)
176. Luenberger, D.C.: *Microeconomic Theory*. McGraw-Hill, New York (1995)
177. Luenberger, D.G.: *Linear and Nonlinear Programming*. Addison-Wesley, Boston (1984)
178. Luo, Z.Q., Tseng, P.: On the convergence of the coordinate descent method for convex differentiable minimization. *J. Optim. Theory Appl.* **72**, 7–35 (1992)

179. Mangasarian, O.L.: Nonlinear Programming. McGraw-Hill, New York (1969)
180. Mangasarian, O.L.: Machine learning via polyhedral concave minimization. In: Fischer, H., Riedmueller, B., Schaeffler, S. (eds.) Applied Mathematics and Parallel Computing: Festschrift for Klaus Ritter, pp. 175–188. Physica, Heidelberg (1996)
181. Maratos, N.: Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems. Imperial College London (University of London), London (1978)
182. Markowitz, H.: Portfolio selection. *J. Financ.* **7**, 77–91 (1952)
183. McCormick, G.P.: Nonlinear Programming: Theory, Algorithm and Application. Wiley, New York (1983)
184. McKinnon, K.I.: Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM J. Optim.* **9**, 148–158 (1998)
185. McShane, E.J.: The Lagrange multiplier rule. *Am. Math. Month.* **80**, 922–925 (1973)
186. Moré, J.J., Sorensen, D.C.: On the use of directions of negative curvature in a modified Newton method. *Math. Program.* **16**, 1–20 (1979)
187. Moré, J.J., Thuente, D.J.: Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Softw.* **20**, 286–307 (1994)
188. Nash, S.G.: A survey of truncated-Newton methods. *J. Comput. Appl. Math.* **124**, 45–59 (2000)
189. Nash, S.G., Sofer, A.: Assessing a search direction within a truncated-Newton method. *Oper. Res. Lett.* **9**, 219–221 (1990)
190. Nelder, J.A., Mead, R.: A simplex method for function minimization. *Comput. J.* **8**, 308–313 (1965)
191. Nemhauser, G.L., Wolsey, L.A.: Integer and Combinatorial Optimization. Wiley, New York (1988)
192. Nesterov, Y., Nemirovskii, A.: Interior-point polynomial algorithms in convex programming. SIAM, Philadelphia (1994)
193. Nesterov, Y.E.: Lectures on Convex Optimization, 2nd edn. Springer, Cham (2018)
194. Nesterov, Y.E.: A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. *Dokl. Akad. Nauk SSSR* **269**, 543–547 (1983)
195. Nocedal, J.: Updating quasi-Newton matrices with limited storage. *Math. Comput.* **35**, 773–782 (1980)
196. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer, New York (2006)
197. Oren, S.S., Spedicato, E.: Optimal conditioning of self-scaling variable metric algorithms. *Math. Program.* **10**, 70–90 (1976)
198. Oren, S.S.: On the selection of parameters in self scaling variable metric algorithms. *Math. Program.* **7**, 351–367 (1974)
199. Ortega, J.M.: Stability of difference equations and convergence of iterative processes. *SIAM J. Numer. Anal.* **10**, 268–282 (1973)
200. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
201. Panier, E.R., Tits, A.L.: Avoiding the maratos effect by means of a nonmonotone line search I. general constrained problems. *SIAM J. Numer. Anal.* **28**(4), 1183–1195 (1991)
202. Petrosyan, L.A., Zenchevich, N.A.: Game Theory, 2nd edn. World Scientific, Singapore (2016)
203. Petrova, S., Solov'ev, A.D.: The origin of the method of steepest descent. *Historia Mat.* **24**, 361–375 (1997)
204. Piccialli, V., Sciandrone, M.: Nonlinear optimization and support vector machines. *Ann. Oper. Res.* **314**, 15–47 (2022)
205. Pinter, J.D.: Global Optimization in Action: Continuous and Lipschitz Optimization. Springer, Dordrecht (1996)
206. Polak, E.: Optimization. Springer, New York (1997)
207. Polak, E., Ribière, G.: Notes sur la convergence de méthodes de directions conjugées. *Rev. Française Informat. Recherche Oper.* **16**, 35–43 (1969)
208. Polyak, B.T.: Introduction to Optimization. Optimization Software, New York (1987)

209. Polyak, T.: The conjugate gradient method in extremum problems. *USSR Comput. Math. Math. Phys.* **9**, 94–112 (1969)
210. Powell, M.J.D.: Convergence properties of algorithms for nonlinear optimization. *SIAM Rev.* **28**(4), 487–500 (1986)
211. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**, 155–162 (1964)
212. Powell, M.J.D.: A method for nonlinear constraints in minimization problems. In: Fletcher, R. (ed.) *Optimization*, pp. 283–298. Academic Press, New York (1969)
213. Powell, M.J.D.: A new algorithm for unconstrained optimization. In: Rosen, J.B., et al. (ed.) *Nonlinear Programming*, pp. 31–65. Academic Press, New York (1970)
214. Powell, M.J.D.: On search directions for minimization algorithms. *Math. Program.* **4**, 193–201 (1973)
215. Powell, M.J.D.: Some global convergence properties of a variable metric algorithm for minimizing without exact line searches. In: Cottle, R.W., Lemke, C.E. (eds.) *Nonlinear Programming*, SIAM-AMS Proceedings, vol. IX, pp. 53–72. SIAM publications, Philadelphia (1976)
216. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: *Numerical Analysis*, pp. 144–157. Springer, Berlin (1978)
217. Powell, M.J.D.: Nonconvex minimization calculations and the conjugate gradient method. *Lect. Notes Math.* **1066**, 122–141 (1984)
218. Powell, M.J.D.: *Approximation Theory and Methods*. Cambridge University Press, Cambridge (1997)
219. Powell, M.J.D.: UOBYQA: unconstrained optimization by quadratic approximation. *Math. Program. Ser. B* **92**, 555–582 (2002)
220. Powell, M.J.D., Yuan, Y.: A recursive quadratic programming algorithm that uses differentiable exact penalty functions. *Math. Program.* **35**, 265–278 (1986)
221. Pshenichny, B.N., Danilin, Y.M.: *Numerical Methods in Extremal Problems*. MIR Publishers, Moscow (1978)
222. R. Pytlak, *Conjugate Gradient Algorithms in Nonconvex Optimization*. Springer, Berlin (2009)
223. M. Raydan, On the Barzilai and Borwein choice of the steplength for the gradient method. *IMA J. Numer. Anal.* **13**, 618–622 (1993)
224. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **7**, 26–33 (1997)
225. Rinaldi, F., Schoen, F., Sciandrone, M.: Concave programming for minimizing the zero-norm over polyhedral sets. *Comput. Optim. Appl.* **46**, 467–486 (2010)
226. Robinson, S.M.: A quadratically-convergent algorithm for general nonlinear programming problems. *Math. Program.* **3**, 145–156 (1972)
227. Rockafellar, R.T., Wets, R.J.B.: *Variational Analysis*. Springer, Heidelberg (1998)
228. Rockafellar, R.T.: Penalty methods and augmented lagrangians in nonlinear programming, in *IFIP Technical Conference on Optimization Techniques*, pp. 418–425. Springer, Berlin (1973)
229. Rockafellar, R.T.: *Convex Analysis*. Princeton University Press, Princeton (2015)
230. Rosenbrock, H.H.: An automatic method for finding the greatest or the least value of a function. *Comput. J.* **3**, 175–184 (1960)
231. Ruszczynski, A.: *Nonlinear Optimization*. Princeton University Press, Princeton (2006)
232. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia (2003)
233. Sawaragi, Y., Nakayama, H., Tanino, T.: *Theory of Multiobjective Optimization*. Elsevier, Amsterdam (1985)
234. Schneider, J.J., Kirkpatrick, S.: *Stochastic Optimization*. Springer, Berlin (2007)
235. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley, New York (1998)
236. Schubert, L.K.: Modification of a quasi-newton method for nonlinear equations with a sparse Jacobian. *Math. Comput.* **24**(109), 27–30 (1970)
237. Sciandrone, M., Placidi, G., Testa, L., Sotgiu, A.: Compact low field mri magnet: design and optimization. *Rev. Sci. Instrum.* **71**, 1534–1538 (2000)

238. Shah, B., Buehler, R., Kemphorne, O.: Some algorithms for minimizing a function of several variables. *J. Soc. Ind. Appl. Math.* **12**, 74–92 (1964)
239. Shamanskii, V.E.: On a modification of Newton's method. *Ukrainskyi Matematichnyi Zhurnal* **19**, 133–138 (1967)
240. Shor, N.Z.: The Subgradient Method, pp. 22–47. Springer, Berlin (1985)
241. Sonnevend, G.: An “analytical centre” for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In: System Modelling and Optimization, pp. 866–875. Springer, Berlin (1986)
242. Spedicato, E., Zhao, J.: Explicit general solution of the Quasi-Newton equation with sparsity and symmetry. *Optim. Methods Softw.* **2**(3–4), 311–319 (1993)
243. Sra, S., Nowozin, S., Wright, S.J.: Optimization for Machine Learning. MIT Press, Cambridge (2012)
244. Stancu-Minasian, I.M.: Fractional Programming: Theory, Methods and Applications, vol. 409. Springer, Berlin (2012)
245. Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.* **20**, 626–637 (1983)
246. Sun, W., Yuan, Y.: Optimization Theory and Methods. Science+Business Media, LLC, New York (2006)
247. Tardella, F.: The fundamental theorem of linear programming: extensions and applications. *Optimization* **60**, 283–301 (2011)
248. Tax, D.M.J., Duin, R.P.W.: Support vector data description. *Mach. Learn.* **54**, 45–66 (2004)
249. Toint, P.L.: An assessment of nonmonotone linesearch techniques for unconstrained optimization. *SIAM J. Sci. Comput.* **17**(3), 725–739 (1996)
250. Toint, P.L.: A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. *Math. Program.* **77**, 69–94 (1997)
251. Torczon, V.: On the convergence of pattern search algorithms. *SIAM J. Optim.* **7**, 1–25 (1997)
252. Tseng, P.: Fortified-descent simplicial search method: a general approach. *SIAM J. Optim.* **10**, 269–288 (1999)
253. Vainberg, M.: Variational Methods for the Study of Nonlinear Operators. Holden-Day, San Francisco (1964)
254. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
255. Weston, J., Elisseeff, A., Scholkopf, B.: Use of the zero-norm with linear models and kernel model. *J. Mach. Learn. Res.* **3**, 1439–1461 (2003)
256. Williams, H.P.: Model Building in Mathematical Programming. Wiley, Chichester (2013)
257. Wolfe, P.: Convergence conditions for ascent methods. *SIAM Rev.* **11**, 226–235 (1969)
258. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming: Theory, Algorithms, and Applications, vol. 27. Springer, Berlin (2012)
259. Wright, S., Recht, B.: Optimization for Data Analysis. Cambridge University Press, Cambridge (2022)
260. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM, Philadelphia (1997)
261. Yuan, Y.X.: A new stepsize for the steepest descent method. *J. Comput. Math.* **24**, 149–156 (2006)
262. Zadeh, N.: A note on the cyclic coordinate ascent method. *Manage. Sci.* **16**, 642–644 (1970)
263. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.* **14**(4), 1043–1056 (2004)

Index

A

- Acceptance of unit step-size, 285
- Accumulation point, 648
- Affine scaling methods, 507
- Angle between two vectors, 55, 650
- Angle condition, 183
- Armijo's method, 191
 - convergence, 194
 - termination, 193
- Artificial variables, 48
- Augmented Lagrangian functions, 452

B

- Backpropagation method, 234
- Ball, 646
- Barrier function, 498
- Barrier methods, 497
- Barzilai-Borwein gradient method, 243, 573
- Block descent methods, 609
- Boundary, 647
- Bounded set, 648
- Box constraints, 74

C

- Carathéodory's Theorem, 688
- Cauchy-Schwarz inequality, 650
- Central path, 509
- Cholesky factorization, 292
 - modified, 293
- Classification of unconstrained algorithms, 180
- Closed set, 646
- Closure of a set, 646
- Coercive function, 40

Coercive function on a set, 45

Compact set, 648

Complexity analysis, 174

- of the steepest descent method, 239

Concave programming, 29, 437

Conical combination of vectors, 399

Conjugate direction method, 249

- finite convergence, 251

Conjugate directions, 249

Conjugate gradient method, 245

- formulae

- Day-Yuan, 263

- Fletcher, 263

- Fletcher-Reeves, 263

- Hestenes-Stiefel, 263

- Liu-Storey, 263

- Polyak-Polak-Ribiére, 263

- for linear least squares, 257

- non quadratic case, 262, 263

- quadratic case

- algorithm, 256

- convergence rate, 258

- finite convergence, 254

- preconditioning, 259

Constraint qualification, 93

- concavity of active constraints, 94

- Kuhn-Tucker, 148

- linear independence of active constraint,

94

- Mangasarian-Fromovitz, 94

- Slater's condition, 95

Contour, 37

Convergence

- global, 171

- local, 171

- Q -linear, 173
- Q -quadratic, 173
- Q -superlinear, 173
- R -linear, 173
- R -superlinear, 173
- Convergence conditions, 182, 184
- Convergence of the PPR method in the Convex case, 266
- Convergence rate, 172
 - gradient method, 235
- Convergence to critical points, 170
- Convergence to stationary points, 179
- Convex
 - function, 689
 - hull, 687
 - set, 681
- Convexity, 681
 - of differentiable functions, 695
 - generalized, 703
- Convexity and concavity at a point, 93
- Convex programming, 24
- Coordinate direction methods, 385
 - methods with simple decrease, 386
- Critical point, 53, 415
- Cubic regularization, 316

- D**
- Davidon, W.C., 336
- Decomposition methods, 589
- Derivative-free linesearch-based methods, 396
- Derivative-free linesearch with positive steps, 402
- Derivative-free method for box constrained problems, 425
- Derivative-free methods, 383
- Derivatives of composite function, 671
- Descent condition
 - first order, 54
 - second order, 56
- Descent direction, 54
- Descent lemma, 232
- Differentiable function, 664
- Directional derivative, 663
- Direction of negative curvature, 55
- Direct search methods, 383
- Distance (or metric), 645
- Dual problem, 119

- E**
- Eigenvalue, 654
- Eigenvector, 654
- Equality-constrained QP, 485

- F**
- Farkas Lemma, 157
- Feasibility problem, 37, 47
- Feasible directions, 69, 70
 - box constraints, 74
 - convex feasible set, 76
 - inequality constraints, 138
 - linear constraints, 71, 73
- Finite-difference methods, 383
- Fletcher-Reeves method, 265
- Forcing function, 182
- Fractional programming, 33
- Frank-Wolfe method (conditional gradient method), 419
- Fréchet differentiability, 664
- Frobenius norm, 652
- Function
 - continuously differentiable, 667
 - twice differentiable, 667

- G**
- Gauss-Newton method, 350
 - globally convergent modification, 353
- Gauss-Seidel method, 595
- Gauss-Southwell method, 615
- Generalized convexity, 26
- Globalization of coordinate and Hooke-Jeeves methods through line searches, 404
- Globally convergent Newton-type algorithm, 283
- Goal programming, 34
- Goldstein conditions, 198
- Gradient
 - vector, 665
- Gradient method, 229, 230
 - with constant step-size, 232
 - finite convergence in the quadratic case, 238
- Gradient method with error, 361
- Gradient projection method, 421

- H**
- Heavy ball method, 244

Hooke-Jeeves method, 392
 Hybrid methods, 284
 Hybrid Newton-type algorithm, 289

I

Implicit filtering, 408
 Incremental methods, 358, 360
 Indefinite matrix, 658
 Inequality-constrained QP, 485
 Inner product, 649
 Interior point methods, 497

- for linear programming, 502
- for nonconvex problems, 522
- for quadratic programming, 521

J

Jacobian matrix, 665
 Jacobi method, 624

K

Kalman filter, 345, 358
 KKT conditions for linearly constrained problems, 109

- box constraints, 111
- linear programming, 114
- non negativity constraints, 110
- quadratic programming, 113
- simplex constraints, 112

L

Lagrangian

- function, 88
- generalized multipliers, 88
- multiplier rule, 88

Lagrangian duality, 121

Least squares problems, 345, 347

Level set, 37

Levenberg-Marquardt method, 350

Limited-memory BFGS (L-BFGS), 378

Limit of a sequence, 647

Limit points, 648

- existence, 169
- uniqueness, 170

Limit points of unconstrained algorithms, 178

Linear combination, 639

Linear dependence and independence of vectors, 640

Linear hull, 640

Linear least squares problem, 42, 63

Linear regression problem, 63

Linear space, 639
 Linear subspace, 639
 Line search

- along a feasible direction, 415
- initial interval, 217
- initial tentative step-size, 217
- safeguarded interpolation, 219
- stopping criteria and failures, 225

Line search algorithm based on strong Wolfe conditions, 215

Line search algorithm based on weak Wolfe conditions, 213

Line search methods, 187

- backtracking methods, 191
- classification, 190
- derivative-free, 204
- derivative-free bidirectional search, 205
- inexact, 190
- optimal step-size, 189

Long step path-following methods, 517

M

Maratos effect, 494

Matrix

- Hessian, 667
- positive (negative) definite , 658
- positive (negative) semidefinite, 658

Minimum norm, 657

Minimum point

- global, 21
- local, 23
- strict global, 22
- strict local, 23
- unconstrained, 23
- uniqueness, 26

Model-based methods, 383, 409

Modification methods, 284, 291

Monotonicity conditions, 700

Multi-objective optimization, 35

N

Nelder-Mead (simplex) method, 394

Nesterov's accelerated gradient method, 246

Newton's method, 275

- globalization using trust region methods, 311
- local convergence and convergence rate, 277

Non homogeneous Farkas Lemma, 162

Nonlinear equations

- Broyden's method, 342
- inexact Newton method, 341

- Newton-type methods, 339
 - residual based methods, 344
 - Nonmonotone
 - Armijo-Goldstein line searches, 541
 - Armijo-type line searches, 538
 - Barzilai-Borwein gradient method, 578
 - derivative-free line searches, 546
 - methods for nonlinear equations, 560
 - Newton's method, 557
 - Nonmonotone methods, 529
 - Norm
 - of a matrix, 652
 - of a vector, 644
 - Normal equations, 63
- O**
- Open set, 646
 - Optimality conditions
 - constrained
 - along feasible directions, 70
 - convex feasible set, 77
 - convex problem, 77
 - first order necessary, 71
 - Fritz John, 88, 147
 - inequality constraints, 140
 - Karush-Kuhn-Tucker, 74, 95
 - Lagrange multiplier rule, 99
 - second order necessary, 71, 102
 - second order sufficient, 104
 - second order sufficient (semi) strong, 107
 - second order sufficient with strict complementarity, 108
 - sufficiency of KKT conditions, 100
 - weak second order sufficient, 107
 - unconstrained
 - convex case, 61
 - convex quadratic function, 62
 - first order necessary, 57
 - second order necessary, 57
 - second order sufficient, 58
 - Optimization algorithm, 167
 - Orthogonal matrices, 654
 - Orthogonal vectors, 651
- P**
- Parallel tangent method (PARTAN), 245
 - Pareto efficient solution, 35
 - Partial derivative, 664
 - Path-following methods, 507
 - Penalty functions, 444
 - Perfect line search, 332
- Polyak-Polak-Ribi  re (PPR) method, 266
 - Polyhedron, 684
 - Positive basis, 401
 - Potential function, 498
 - Potential reduction methods, 508
 - PPR algorithm in the non quadratic case, 269
 - Primal-dual path-following methods, 513
 - Projected spectral gradient method, 585
 - Projection, 414
 - Projection on a convex set, 81
 - box constraints, 83
 - characterization, 81
 - nonnegativity constraints, 83
 - optimality conditions, 85
 - Proximal-point, 607
 - Pseudoconvex function, 706
 - Pseudoinverse matrix, 63
- Q**
- Quadratic form, 658
 - Quasi-convex function, 703
 - Quasi-Newton
 - BFGS algorithm, 333
 - BFGS method, 332
 - convergence analysis, 335
 - global convergence, 334
 - Broyden class, 331
 - Broyden's method, 329
 - DFP method, 330
 - equations, 326
 - limited-memory, 376
 - memoryless, 376
 - methods, 325
 - rank one formulae, 328
 - rank two formulae, 329
 - Quasi-Newton methods
 - finite termination in the quadratic case, 332
- R**
- Rayleigh quotients, 575
 - Real matrices, 651
- S**
- Saddle point, 60, 125
 - Scalar product, 649
 - Second order derivatives, 667
 - Sequential Minimal Optimization (SMO) algorithms, 627
 - Sequential penalty algorithm, 448

Sequential quadratic programming (SQP)
 methods, 481
Shamanskii method, 280
Sherman-Morrison-Woodbury formula, 327
Short step path-following methods, 517
Simplex gradient, 406
Singular value decomposition, 656
Spacer steps, 231
Sparse optimization, 434
Spectral decomposition, 655
Spectral gradient methods, 243, 573
Spectral gradient methods for nonlinear
 equations, 580
Starting point, 168
Stationary point, 57
Steepest descent direction, 229
Strong duality, 121, 124
Strongly convex function, 703
Support Vector Machine (SVM), 130
Sylvester criterion, 660
Symmetric matrix, 654
Systems of linear equalities, 641

T

Tangent direction, 141
 - basic optimality condition, 141
 - characterization, 144
 - optimality condition: equality and
 inequality constraints, 146
Termination criterion, 168

Theorem

- Fredholm, 154
- mean value, 669
- Motzkin, 160
- Ostrowski, 170
- of Taylor, 670
- Weierstrass, 37

Theorems of the alternative, 137

Trace, 651

Truncated Newton method, 366

Trust region methods, 284, 297

- algorithm model, 299
- Cauchy step method, 302
- conjugate gradient method of Steihaug,
 306
- convergence results, 300, 301
- dogleg method, 304
- necessary and sufficient optimality
 conditions, 308
- sufficient reduction, 300

W

Watchdog techniques, 552

Weak duality, 119, 122

Wolfe conditions, 201

Wolfe's dual, 126

Z

Zero-norm, 434