

# **TRUST-REGION METHODS**



## MPS/SIAM Series on Optimization

This series is published jointly by the Mathematical Programming Society and the Society for Industrial and Applied Mathematics. It includes research monographs, textbooks at all levels, books on applications, and tutorials. Besides being of high scientific quality, books in the series must advance the understanding and practice of optimization and be written clearly, in a manner appropriate to their level.

**Editor-in-Chief**

John E. Dennis, Jr., Rice University

**Continuous Optimization Editor**

Stephen J. Wright, Argonne National Laboratory

**Discrete Optimization Editor**

David B. Shmoys, Cornell University

**Editorial Board**

Daniel Bienstock, Columbia University

John R. Birge, Northwestern University

Andrew V. Goldberg, InterTrust Technologies Corporation

Matthias Heinkenschloss, Rice University

David S. Johnson, AT&T Labs - Research

Gil Kalai, Hebrew University

Ravi Kannan, Yale University

C. T. Kelley, North Carolina State University

Jan Karel Lenstra, Technische Universiteit Eindhoven

Adrian S. Lewis, University of Waterloo

Daniel Ralph, The Judge Institute of Management Studies

James Renegar, Cornell University

Alexander Schrijver, CWI, The Netherlands

David P. Williamson, IBM T.J. Watson Research Center

Jochem Zowe, University of Erlangen-Nuremberg, Germany

**Series Volumes**

Conn, Andrew R., Gould, Nicholas I. M., and Toint, Philippe L., *Trust-Region Methods*

# TRUST-REGION METHODS

Andrew R. Conn

IBM-Thomas J. Watson Research Center  
Yorktown Heights, New York, United States

Nicholas I. M. Gould  
Rutherford Appleton Laboratory  
Chilton, Oxfordshire, England

Philippe L. Toint  
University of Namur  
Namur, Belgium

**siam.**

Society for Industrial and Applied Mathematics  
Philadelphia

**MPS**

Mathematical Programming Society  
Philadelphia

Copyright © 2000 by Society for Industrial and Applied Mathematics and Mathematical Programming Society.

10 9 8 7 6 5 4 3 2 1

All rights reserved. Printed in the United States of America. No part of this book may be reproduced, stored, or transmitted in any manner without the written permission of the publisher. For information, write to the Society for Industrial and Applied Mathematics, 3600 University City Science Center, Philadelphia, PA 19104-2688.

#### **Library of Congress Cataloging-in-Publication Data**

Conn, A. R. (Andrew R.)

Trust-region methods / Andrew R. Conn, Nicholas I. M. Gould, Philippe L. Toint.  
p. cm. — (MPS-SIAM series on optimization). Includes bibliographical references and index.

ISBN 0-89871-460-5

1. Mathematical optimization. I. Gould, Nicholas, I. M. II. Toint, Ph. L. (Philippe L.)

III. Title. IV. Series.

QA402.5 .C6485 2000

519.3—dc21

00-038796

#### **About the cover**

Cover art © Rose-Marie Warzée, Oil on Canvas, 80x100 cm, Title: *Cape Point*, at the Contemporary Art Gallery Rive Gauche, Avenue Baron Huart, 28, B-5000 Namur, Belgium. Photographer: Jean-Luc Tillière.

Rose-Marie Warzée is a Belgian artist who studied art and screen printing in Brussels. She now paints with oil on canvas. The many layers of color she uses in a transparent manner are plainly visible when viewing her paintings. The basic construction rests on the square or the circle elements which disappear little by little and invite one to escape. The circle becomes a whirl and the square opens itself to the outside. *Cape Point* is a circle which represents the globe and is an osmose between the earth, the blue sky, and the two oceans (reference at South Africa).

Cover art reprinted with permission from Jean-Luc Tillière.

**siam** is a registered trademark.

To

Barbara, Leah, and Jeremy

Penny Gould and Hilary Morrison

Claire Manil, Lucie Leclercq, and Jacques Toint

with love and thanks

# Contents

<b>Preface</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What Is a Trust-Region Algorithm? . . . . .	1
1.2 Historical Development of Trust-Region Methods . . . . .	8
1.3 Some Applications of Trust-Region Algorithms . . . . .	9
<b>I Preliminaries</b>	<b>13</b>
<b>2 Basic Concepts</b>	<b>15</b>
2.1 Basic Notation . . . . .	15
2.2 Eigenvalues and Eigenvectors . . . . .	16
2.3 Vector and Matrix Norms . . . . .	20
2.3.1 Vector and Functional Norms . . . . .	21
2.3.2 Matrix Norms . . . . .	22
<b>3 Basic Analysis and Optimality Conditions</b>	<b>25</b>
3.1 Basic Real Analysis . . . . .	25
3.1.1 Basic Topology . . . . .	25
3.1.2 Derivatives and Taylor's Theorem . . . . .	27
3.1.3 Convexity . . . . .	30
3.1.4 Nonsmooth Functions . . . . .	33
3.1.5 Geometry . . . . .	34
3.2 Optimality Conditions . . . . .	37
3.2.1 Differentiable Unconstrained Problems . . . . .	38
3.2.2 Differentiable Constrained Problems . . . . .	39
3.2.3 Convex Programming . . . . .	44
3.2.4 Nonsmooth Problems . . . . .	47
3.3 Sequences and Basic Iterations . . . . .	50
3.3.1 Sequences . . . . .	50
3.3.2 Newton's Method . . . . .	51
3.3.3 Forcing Functions . . . . .	53

<b>4 Basic Linear Algebra</b>	<b>55</b>
4.1 Linear Equations in Optimization . . . . .	55
4.1.1 Structure and Sparsity . . . . .	56
4.1.2 Structured Systems . . . . .	57
4.2 Special Matrices . . . . .	57
4.2.1 Diagonal Matrices . . . . .	58
4.2.2 Triangular Matrices . . . . .	58
4.2.3 Orthonormal Matrices . . . . .	58
4.2.4 Tridiagonal and Band Matrices . . . . .	59
4.3 Direct Methods for Solving Linear Systems . . . . .	60
4.3.1 Stability in the Face of Rounding Errors . . . . .	60
4.3.2 Sparse Systems . . . . .	61
4.3.3 Symmetric Positive Definite Matrices . . . . .	63
4.3.4 Symmetric Indefinite Matrices . . . . .	65
4.3.5 Frontal and Multifrontal Methods . . . . .	66
4.3.6 Matrix Modification . . . . .	67
4.4 Least-Squares Problems and Projections . . . . .	68
4.4.1 Least-Squares Problems . . . . .	68
4.4.2 Projections, Range-, and Null-Space Bases . . . . .	71
<b>5 Krylov Subspace Methods</b>	<b>75</b>
5.1 The Conjugate Gradient Method . . . . .	75
5.1.1 Minimization in a Subspace . . . . .	76
5.1.2 Conjugate Directions . . . . .	77
5.1.3 Generating Conjugate Directions . . . . .	78
5.1.4 Conjugate Gradients . . . . .	82
5.1.5 Convergence of the Conjugate Gradient Method . . . . .	83
5.1.6 Preconditioned Conjugate Gradients . . . . .	86
5.2 The Lanczos Method . . . . .	91
5.2.1 Computing an Orthonormal Basis for the Krylov Space . . . . .	92
5.2.2 Relationship with the Conjugate Direction Method . . . . .	95
5.2.3 Finding Conjugate Directions from an Orthonormal Krylov Basis . . . . .	96
5.2.4 Approximation of Critical Points within the Subspace . . . . .	99
5.2.5 Rayleigh–Ritz Approximations to Eigenpairs . . . . .	101
5.2.6 Preconditioned Lanczos . . . . .	102
5.3 Linear Least-Squares Problems . . . . .	106
5.4 Problems with Constraints . . . . .	108
5.4.1 Projected Preconditioned Conjugate Gradients . . . . .	109

<b>II Trust-Region Methods for Unconstrained Optimization</b>	<b>113</b>
<b>6 Global Convergence of the Basic Algorithm</b>	<b>115</b>
6.1 The Basic Trust-Region Algorithm . . . . .	115
6.2 Assumptions . . . . .	121
6.2.1 Assumptions on the Problem . . . . .	121
6.2.2 Assumptions on the Algorithm . . . . .	122
6.3 The Cauchy Point and the Model Decrease . . . . .	123
6.3.1 The Cauchy Arc . . . . .	123
6.3.2 The Cauchy Point for Quadratic Models . . . . .	124
6.3.3 The Approximate Cauchy Point . . . . .	128
6.3.4 The Final Condition on the Model Decrease . . . . .	130
6.4 Convergence to First-Order Critical Points . . . . .	133
6.5 Second-Order Convex Models . . . . .	139
6.6 The Eigenpoint and Second-Order Nonconvex Models . . . . .	147
6.6.1 Exploitation of Negative Curvature by Minimization . . . . .	148
6.6.2 Exploitation of Negative Curvature by a Linesearch . . . . .	150
6.6.3 Convergence Theorems . . . . .	153
6.7 Trust-Region Scaling . . . . .	162
6.7.1 Geometry and Scaling . . . . .	162
6.7.2 Uniformly Equivalent Norms Again . . . . .	166
<b>7 The Trust-Region Subproblem</b>	<b>169</b>
7.1 The Solution of Trust-Region Subproblems . . . . .	169
7.2 Characterization of the $\ell_2$ -Norm Model Minimizer . . . . .	171
7.3 Finding the $\ell_2$ -Norm Model Minimizer . . . . .	176
7.3.1 Finding the $\ell_2$ -Norm Model Minimizer . . . . .	176
7.3.2 Finding the Root of $\ s(\lambda)\ _2 - \Delta = 0$ . . . . .	181
7.3.3 Newton's Method and the Secular Equation . . . . .	182
7.3.4 Safeguarding Newton's Method . . . . .	185
7.3.5 Updating the Intervals of Uncertainty . . . . .	189
7.3.6 Finding $\lambda$ in the Interval of Uncertainty . . . . .	189
7.3.7 Finding Good Lower Bounds on $-\lambda_1$ . . . . .	190
7.3.8 Initial Values . . . . .	192
7.3.9 The Complete Algorithm . . . . .	193
7.3.10 Termination . . . . .	194
7.3.11 Enhancements . . . . .	197
7.4 The Scaled $\ell_2$ -Norm Problem . . . . .	200
7.5 Approximating the Model Minimizer . . . . .	201
7.5.1 The Truncated Conjugate Gradient Method . . . . .	202
7.5.2 How Good Is the Steihaug–Toint Point? . . . . .	208
7.5.3 Dogleg and Double-Dogleg Paths . . . . .	218
7.5.4 The Truncated Lanczos Approach . . . . .	221

7.5.5	Computing the Eigenpoint . . . . .	230
7.5.6	Eigenvalue-Based Approaches . . . . .	231
7.6	Projection-Based Approaches . . . . .	235
7.7	Norms that Reflect the Underlying Geometry . . . . .	236
7.7.1	The Ideal Trust Region . . . . .	236
7.7.2	The Absolute-Value Trust Region . . . . .	237
7.7.3	Solving Diagonal and Block Diagonal Trust-Region Subproblems . . . . .	238
7.7.4	Coping with Singularity . . . . .	239
7.8	The $\ell_\infty$ -Norm Problem . . . . .	243
<b>8</b>	<b>Further Convergence Theory Issues</b>	<b>249</b>
8.1	Convergence of General Measures . . . . .	249
8.1.1	First-Order and Criticality Measures . . . . .	249
8.1.2	First-Order Convergence Theory . . . . .	251
8.1.3	Second-Order Convergence Theory . . . . .	255
8.1.4	Convergence in Dual Norms . . . . .	259
8.1.5	A More General Cauchy Point . . . . .	265
8.1.6	The Scaled Cauchy Point . . . . .	269
8.2	Weakly Equivalent Norms . . . . .	272
8.3	Minimization in Infinite-Dimensional Spaces . . . . .	274
8.3.1	Hilbert Spaces . . . . .	275
8.3.2	Banach Spaces . . . . .	277
8.4	Using Approximate Derivatives . . . . .	280
8.4.1	Concepts and Assumptions . . . . .	280
8.4.2	Global Convergence . . . . .	285
8.4.3	Finite-Difference Approximations to Derivatives . . . . .	297
8.5	Composite Problems and Models . . . . .	302
<b>9</b>	<b>Conditional Models</b>	<b>307</b>
9.1	Motivation and Formal Description . . . . .	307
9.2	Convergence to First-Order Critical Points . . . . .	313
9.3	Convergence to Second-Order Critical Points . . . . .	319
9.4	Conditional Models and Derivative-Free Optimization . . . . .	322
9.4.1	Derivative-Free Minimization: Why and How? . . . . .	322
9.4.2	Basic Concepts in Multivariate Interpolation . . . . .	324
9.4.3	The Interpolation Error . . . . .	330
9.5	Conditional Models and Models with Memory . . . . .	336
9.5.1	Adding Memory to the Model of the Objective Function . . . . .	336
9.5.2	The Effect of Memory on the Modelling Error . . . . .	338

<b>10 Algorithmic Extensions</b>	<b>347</b>
10.1 A Nonmonotone Trust-Region Algorithm . . . . .	347
10.1.1 The Nonmonotone Algorithm . . . . .	348
10.1.2 Convergence Theory Revisited . . . . .	350
10.1.3 The Reference Iteration and Other Practicalities . . . . .	355
10.2 Structured Trust Regions . . . . .	359
10.2.1 Motivation and Basic Concepts . . . . .	359
10.2.2 A Trust-Region Algorithm Using Problem Structure . . . . .	364
10.2.3 Convergence Theory . . . . .	370
10.3 Trust Regions and Linesearches . . . . .	376
10.3.1 Linesearch Algorithms as Trust-Region Methods . . . . .	376
10.3.2 Backtracking at Unsuccessful Iterations . . . . .	380
10.3.3 Which Steps Are Gradient Related? . . . . .	383
10.4 Other Measures of Model Accuracy . . . . .	387
10.4.1 Magical Steps . . . . .	387
10.4.2 Correction Steps . . . . .	391
10.4.3 Modified Models . . . . .	392
10.5 Alternative Trust-Region Management . . . . .	394
10.5.1 Internal Doubling . . . . .	394
10.5.2 Basing the Radius Update on the Steplength . . . . .	396
10.6 Problems with Dynamic Accuracy . . . . .	399
10.6.1 An Algorithm Using Dynamic Accuracy . . . . .	400
10.6.2 Conditional Models and Dynamic Accuracy . . . . .	406
<b>11 Nonsmooth Problems</b>	<b>407</b>
11.1 Algorithms for Nonsmooth Optimization . . . . .	408
11.2 Convergence to a First-Order Critical Point . . . . .	414
11.3 Variations on This Theme . . . . .	423
11.3.1 A Nonmonotonic Variant . . . . .	423
11.3.2 Correction Steps . . . . .	424
11.4 Computing the Generalized Gradient . . . . .	426
11.5 Suitable Models . . . . .	434
<b>III Trust-Region Methods for Constrained Optimization with Convex Constraints</b>	<b>439</b>
<b>12 Projection Methods for Convex Constraints</b>	<b>441</b>
12.1 Simple Feasible Domains and Their Geometry . . . . .	441
12.1.1 Simple Bounds on the Variables . . . . .	442
12.1.2 Other Simple Domains . . . . .	443
12.1.3 The Projected-Gradient Path . . . . .	444
12.1.4 A New Criticality Measure . . . . .	448

12.2	An Algorithm for Problems with Convex Constraints . . . . .	451
12.2.1	The Generalized Cauchy Point . . . . .	453
12.2.2	Convergence to First-Order Critical Points . . . . .	458
12.3	Active Constraints Identification . . . . .	460
12.3.1	Further Assumptions . . . . .	460
12.3.2	The Geometry of the Set of Limit Points . . . . .	461
12.3.3	The Identification of Active Constraints . . . . .	467
12.4	Convergence to Second-Order Critical Points . . . . .	474
12.4.1	The Role of the Lagrangian . . . . .	474
12.4.2	Convex Models . . . . .	476
12.4.3	Nonconvex Models . . . . .	481
<b>13</b>	<b>Barrier Methods for Inequality Constraints</b>	<b>491</b>
13.1	Convex Constraints and Barriers . . . . .	491
13.2	A Trust-Region Method for Barrier Functions . . . . .	498
13.3	Constrained Cauchy and Eigenpoints . . . . .	504
13.4	The Primal Log-Barrier Algorithm . . . . .	511
13.5	Reciprocal Barriers . . . . .	517
13.6	A Primal-Dual Algorithm . . . . .	518
13.6.1	The Algorithm . . . . .	518
13.6.2	Convergence Properties . . . . .	522
13.6.3	Updating the Vector of Dual Variables . . . . .	525
13.7	Scaling of Barrier Methods . . . . .	527
13.7.1	Reintroducing Iteration-Dependent Norms . . . . .	527
13.7.2	Scaling of the Inner Iterations . . . . .	528
13.7.3	Scaling of the Outer Iterations . . . . .	530
13.8	Upper and Lower Bounds on the Variables . . . . .	535
13.9	Barrier Methods for General Constraints . . . . .	536
13.10	Adding Linear Equality Constraints . . . . .	542
13.11	The Affine-Scaling Method . . . . .	550
13.12	The Method of Coleman and Li . . . . .	554
13.12.1	The Algorithm . . . . .	554
13.12.2	Convergence Theory . . . . .	559
<b>IV</b>	<b>Trust-Region Methods for General Constrained Optimization and Systems of Nonlinear Equations</b>	<b>571</b>
<b>14</b>	<b>Penalty-Function Methods</b>	<b>573</b>
14.1	Penalty Functions and Constrained Optimization . . . . .	573
14.2	Smooth Penalty Functions . . . . .	575
14.3	Quadratic Penalty-Function Methods . . . . .	582
14.4	Augmented Lagrangian Function Methods . . . . .	593

14.5 Nonsmooth Exact Penalty Functions . . . . .	610
14.6 Smooth Exact Penalty Functions . . . . .	616
<b>15 Sequential Quadratic Programming Methods</b>	<b>623</b>
15.1 Introduction . . . . .	623
15.2 What Is Sequential Quadratic Programming? . . . . .	624
15.2.1 Methods for Problems with Equality Constraints . . . . .	624
15.2.2 Methods for Inequality Constraints . . . . .	631
15.2.3 Quadratic Programming . . . . .	633
15.3 Merit Functions and SQP Methods . . . . .	636
15.3.1 The Augmented Lagrangian Penalty Function . . . . .	637
15.3.2 Nonsmooth Exact Penalty Functions . . . . .	637
15.3.3 Smooth Exact Penalty Functions . . . . .	655
15.4 Composite-Step Trust-Region SQP Methods . . . . .	657
15.4.1 Vardi-Like Approaches . . . . .	658
15.4.2 Byrd-Omojokun-like Approaches . . . . .	694
15.4.3 Celis-Dennis-Tapia-like Approaches . . . . .	711
15.4.4 Inequality Constraints . . . . .	717
15.5 The Filter Method . . . . .	721
15.5.1 A Composite-Step Approximate SQP Framework . . . . .	721
15.5.2 The Notion of a Filter . . . . .	725
15.5.3 An SQP Filter Algorithm . . . . .	727
15.5.4 Convergence to First-Order Critical Points . . . . .	730
15.5.5 An Alternative Step Strategy . . . . .	742
15.6 Nonquadratic Models . . . . .	745
15.7 Concluding Remarks . . . . .	746
<b>16 Nonlinear Equations and Nonlinear Fitting</b>	<b>749</b>
16.1 Nonlinear Equations, Nonlinear Least Squares . . . . .	749
16.2 Nonlinear Equations in Other Norms . . . . .	759
16.2.1 Global Convergence . . . . .	761
16.2.2 Asymptotic Convergence of the Basic Method . . . . .	761
16.2.3 Second-Order Corrections . . . . .	766
16.3 Complementarity Problems . . . . .	770
16.3.1 The Problem and an Associated Merit Function . . . . .	770
16.3.2 Applying the Basic Nonsmooth Algorithm . . . . .	772
16.3.3 Regular Complementarity Problems and Their Solutions . . . . .	774
<b>V Final Considerations</b>	<b>779</b>
<b>17 Practicalities</b>	<b>781</b>
17.1 Choosing the Algorithmic Parameters . . . . .	781
17.2 Choosing the Initial Trust-Region Radius . . . . .	784

17.3 Computing the Generalized Cauchy Point . . . . .	789
17.4 Other Numerical Considerations . . . . .	792
17.4.1 Computing the Model Decrease . . . . .	792
17.4.2 Computing the Value of $\rho_k$ . . . . .	793
17.4.3 Stopping Conditions . . . . .	794
17.4.4 Noise and/or Expensive Function Evaluations . . . . .	798
17.5 Software . . . . .	799
<b>Afterword</b>	<b>801</b>
<b>Appendix: A Summary of Assumptions</b>	<b>803</b>
<b>Annotated Bibliography</b>	<b>813</b>
<b>Subject and Notation Index</b>	<b>935</b>
<b>Author Index</b>	<b>951</b>

# Preface

For many years now, the three of us have been involved in the development and implementation of algorithms for large-scale numerical optimization. We have been surrounded by colleagues in universities, in government, and in industrial research laboratories for whom the solution of such problems is just a small, but often vital, part of their day-to-day activities. For them, the most important or, more exactly, *only* concern about the methods we propose is that they should be reliable: there is no point in developing a complicated model of, say, the economy, or an industrial plant, or a molecule, or even a stellar system only to find that the optimizations required fail simply because the assumptions made by the optimization algorithm do not hold. It is for this reason that we first became interested in trust-region methods. We were attracted by the general applicability, the elegant theory, the simplicity of concepts involved, and, most especially, the success in practice of the resulting methods. Of particular significance is that the methods we implement are normally as close to the algorithms implied by the theory as the floating-point world allows, and this leads to a confidence in the software that both we and our colleagues produce.

As a consequence, an important reason for publishing this book is to collect together in an accessible way our enthusiasm, knowledge, and research in the area while at the same time attempting to give a broad view of the work of our colleagues. In particular, the intention is to present a comprehensive and well-motivated text on trust-region methods. The book is meant to be largely self-contained and is written at the level of a graduate student with a basic background in numerical analysis, real analysis, and numerical linear algebra. Some of the material has already been used in a graduate course on optimization.

Three major aims are, firstly, a detailed description of the basic theory of trust-region methods and the resulting algorithms for unconstrained, linearly constrained, and generally constrained optimization; secondly, the inclusion of implementation and computational details; and finally, substantive material on less well-known advanced topics, including structured trust regions, derivative-free methods, approximate methods (including noise), nonmonotone algorithms, and nonsmooth problems. We have also included a comprehensive list of notated references. Since we are not naive enough to presume that the book will be free of errors (typographical or otherwise), we shall be maintaining (mirrored) WWW pages

<http://www.numerical.rl.ac.uk/trbook/trbook.html>

and

<http://thales.math.fundp.ac.be/pub/trbook/trbook.html>

containing corrections and updates. These pages will also include the complete **BIBTEX**.bib file for the references cited in this book, as well as any subsequent updates. We would be delighted to receive any corrections or updates to these lists.

Once it became known that we were writing this book, one of the most common questions we were asked was what the relationship is between trust-region methods and what are commonly seen as their archrivals, linesearch methods. At first glance, the two approaches are very different. The linesearch approach is a young person's dream: pick any old direction, and charge downhill as fast as possible regardless of the consequences. The trust-region approach, by contrast, is ideally suited to those of us who now look back upon our youth with fond memories: first consider the consequences of making a step by modelling how you think things will turn out, and then have some means of adapting when this model of reality lets you down or exceeds your wildest expectations. In fact, the two approaches are far closer than outsiders might have been led to believe. As we shall see, one can actually view linesearch methods as special cases of trust-region methods. We indeed believe that the two approaches are very close, and that which one chooses to prefer is rather akin to preferring the dangers of bell-ringing to those of volleyball, Bordeaux to Bourgogne, or Messiaen to Mozart—i.e., blissful prejudice. We maintain that the happiest optimizer is the one who is fluent in both approaches.

At the end of our enterprise, we can only hope that the reader will enjoy our book. We have learned more than we thought possible, and, more importantly, feel that there is still far more waiting to be uncovered.

## General Outline of the Book

The book is divided into five parts.

Part I gives the essential background. It is separated into four chapters dealing successively with the basic mathematical concepts, the necessary analytic tools and related optimality conditions, the required linear algebra background, and a more specific chapter on Krylov methods for the solution of linear systems and eigenvalue calculations.

Trust-region methods for unconstrained optimization is the topic of Part II. Chapter 6, the first of this part, describes the basic trust-region algorithm and develops its global convergence theory. We recommend that a reader who is only interested in understanding the basic tools of trust-region analysis should read this chapter directly after the introductory Chapter 1. The next chapter gives a detailed description of how the trust-region subproblems are solved. Extensions of the convergence theory are considered in Chapter 8, including the use of approximate derivative information and the analysis of trust-region methods in infinite-dimensional spaces. A chapter on

conditional models and their application in derivative-free algorithms then follows. We then devote Chapter 10 to further algorithmic extensions of the basic trust-region algorithm that includes nonmonotone algorithms, structured trust regions, methods for composite problems, backtracking techniques, and a discussion of the relation between trust-region and linesearch methods. Finally, there is a chapter on approaches for nonsmooth problems.

Part III considers trust-region methods in the case of problems with convex constraints. It is divided into two chapters, dealing with projection-based approaches (Chapter 12) and barrier interior-point techniques (Chapter 13), respectively.

In Part IV general constrained optimization is addressed. Chapter 14 describes the penalty approach, including augmented Lagrangian methods. Chapter 15 covers approaches based on sequential quadratic programming. Chapter 16 deals with systems of nonlinear equations, least squares, and complementarity problems.

Besides the bibliography, Part V contains a few useful summary sections. Chapter 17 is devoted to practical matters and covers initialization and numerical matters as well as software questions. We offer an afterword, and then we list all assumptions used in the theoretical developments, together with an indication of the pages where they have been defined, in an appendix. Two indices are provided: a subject and notation index and an author index.

## Notational Conventions

Designing consistent notation that is, at the same time, compliant with standard use is one of the nightmares in writing a book like this one. We have tried to be as reasonable as possible in striving for these goals. Globally, we use Greek symbols for scalars, lower case roman for vectors, upper case roman for matrices, and calligraphic for sets. But we have made exceptions to be consistent with established tradition in a few cases. We also tried to provide mnemonic subscripts for useful constants, whenever possible. Constants that are important only in the chapter in which they appear may only be assigned a number.

We have also endeavoured to use a constructive scheme in the labelling of assumptions, although we readily admit that it is sufficiently complex to perhaps not always succeed. We briefly review it here for future reference, without taking care to define all the terms. Firstly, all assumptions begin with the letter A. The second letter indicates the object of the assumption. We use eight such characters, namely F for the objective *function*, M for the *model*, A for the *algorithm*, N for the *norm*, C for the *constraints*, O for the *optimality* conditions, W for the *weighted* composite (i.e., of the objective and constraints) function, and I for the *iterates*. The number that follows the period is meant, when possible, to indicate some horizontal classification over these objects, whereas a lower case letter that optionally follows enables us to identify assumptions that are essentially of the same type. The early numbers for the algorithm, function (i.e., objective, constraints, nonlinear equations), and model classes have special meaning, while for other classes, they do not. In assumptions concerning an algorithm we

use 1 to denote those related to the model decrease and 2 for eigenpoint decrease. For functions, 1 denotes differentiability levels, 2 denotes a lower bound on the function, 3 indicates boundedness of the Hessian, 4 denotes a bound on the third-order tensor, and 5 indicates boundedness of the gradient. Similarly, for models, 1 denotes differentiability levels; 2, 3, and 5 denote agreement with the function, gradient, and Hessian values, respectively; and 4 and 6 denote boundedness and Lipschitz continuity of the model Hessian, respectively. Thus, for example, AF.1 and AC.1 are both assumptions concerning twice-differentiable functions, where the first pertains to the objective function and the second to constraints, whereas AA.1 and AA.1b are both assumptions on the algorithm related to the model decrease. AM.5b relates to the model Hessian agreeing with the function Hessian, in this particular case at a critical point. Finally, in three cases, the last lower case letter has a special meaning: e refers to systems of nonlinear *equations*, s to partially *separable* problems, and n to *nonsmooth* problems, although they are applied consistently with the other conventions.

We encourage the reader to use the index at the end of the book whenever he or she feels lost.

## Acknowledgements

It remains for us to offer our thanks to the many people who have helped us with this book, directly or indirectly, knowingly or in ignorance. We salute them all. Firstly, global thanks are due to

- Barbara, Penny, and Claire, our spouses, whose support, understanding, and patience was crucial during the project;
- Annick Sartenaer, both for her enjoyable collaboration on several of the topics that led to this book and for her uncanny ability to find typos and offer improvements to our manuscript;
- Richard Byrd, Iain Duff, Francisco Facchinei, Michael Ferris, Roger Fletcher, Masao Fukushima, Matthias Heinkenschloss, Christian Kanzow, Jacko Koster, Sven Leyffer, Stefano Lucidi, Ken McKinnon, Karl Meerbergen, Jorge Nocedal, Dominique Orban, Petr Plechac, John Reid, Massimo Roma, Stefan Scholtes, Jennifer Scott, Paul Tseng, Michael Ulbrich, Stefan Ulbrich, Luis Vicente, Joseph Winkin, Henry Wolkowicz, Yin Xiao, and Yaxiang Yuan for the sound technical advice and moral support they offered;
- Rafik Baccari, Sébastien Barette, Fabian Bastin, Myrana Brohé, Benoît Colson, Christelle Denis, Alexandra Dessimoz, Delphine Donfut, David Dumont, Lara Favaro, Marie-Pierre Fivet, Sabine Gillmann, François Glineur, Joël Grad, Stéphane Grad, Anne Hespel, Didier Jacquemin, David Jungblut, Catherine Lebeau, Emmanuelle Lincé, Benoît Masquillier, Thierry Martens, Isabelle Motte, Noëlle Paquay, Valérie Pochon, Monique Polet, Bénédicte Schmeler, and Alain Weis, all

students at the University of Namur, who suffered through a succession of drafts of several of the chapters contained here—for their input, which greatly helped to improve them;

- Murielle Haguinet and Chantal Ippersiel, whose help was precious when compiling the bibliography;
- Rose-Marie Warzée, who accepted our use of one of her paintings for the cover illustration, and the Rive Gauche gallery (Namur) for supporting the idea;
- the developers of L<sup>A</sup>T<sub>E</sub>X, MATLAB, SCILAB, GNUPLOT, Linux, and Xfig, who made typesetting and illustrating this book possible;
- Eric Cornélis, our L<sup>A</sup>T<sub>E</sub>X wizard, who agreed to work at weekends to make sure the package would cope with our ever-increasing demands;
- Beth Gallagher as well as Vickie Kearn, Deborah Poulson, Kelly Thomas, and the crew at SIAM, for their supportive helpfulness; and
- our employers, the IBM T. J. Watson Research Center, the Rutherford Appleton Laboratory, and the University of Namur, as well as our home-from-home at CERFACS in Toulouse.

On a more personal level, Andy would like to thank the many wonderful colleagues with whom he has interacted over the years and without whom his professional life would have been much poorer.

Nick would like to thank Sue Wood, the world's nicest dentist; Paul and Frankie Hexter, and the early-evening crowd at the Royal Oak, Wantage, for the sustenance and general bonhomie; Pat Fish, Robyn Hitchcock, and Bob Mould for the songs; the people of Childrey for their kindness; Paul, Dave, Lindsay, Dave, Susan, George, Sheila, Harry, Jean, Paul, Jane, Derek, Alan, June, Paul, Steve, Judi, Fen, Pauline, and the rest of the Brighton posse for refusing to grow old with us; Nick, Jessie, Imogen, Lauren, Alice, George, and Alice for keeping us young; and Madrabug, Swaysie, Sam, Matt, and Tisa for being best man's best friends.

Philippe would like to thank Cécile and Sarah for their communicative appreciation of life; Kay Axhausen, Juliane and Guy Fontinoy, M. Patriksson, John Polak, Annie and Madeleine Toint, and all members of the Transportation Research Group at the University of Namur for their explicit or implicit encouragements; as well as Claudio Monteverdi, Johann Sebastian Bach, Johannes Brahms, Anton Webern, Olivier Messiaen, Guidon Kremer, Ralph Towner and Oregon, Jan Garbarek, Chick Corea, and Charlie Haden for their music.

ANDREW R. CONN  
NICHOLAS I. M. GOULD  
PHILIPPE L. TOINT

# Chapter 1

---

---

## Introduction

---

---

### 1.1 What Is a Trust-Region Algorithm?

A simple illustration may be the easiest and most intuitive way to answer this question. We therefore assume that we are interested in finding the value of a two-component vector of real *variables*  $x$  such that it minimizes some *objective function*  $f(x)$ . This function may represent the potential energy of some physical system, say, in which case its minimizers might correspond to the system's stable states, or it may describe the cost of using a company management policy described by two parameters. The range of situations in which one wishes to minimize some objective function is indeed very wide, as we will see in Section 1.3. Assume that the objective function we are interested in is given by the formula

$$f([x]_1, [x]_2) = -10[x]_1^2 + 10[x]_2^2 + 4 \sin([x]_1[x]_2) - 2[x]_1 + [x]_1^4,$$

where  $[x]_1$  and  $[x]_2$  are the two components of the vector of variables  $x$ . The contour lines of the function are plotted in Figure 1.1.1.

Of course, if we already know the value of the function everywhere, as is the case when we produce a plot like that of Figure 1.1.1, finding the point  $x_*$  at which  $f(x_*)$  is lowest is not too difficult. But this procedure is considerably more costly than necessary, in that we need to compute the value of our objective function at many (if not all) possible points. Our approach is different: we assume only that we know the value of  $f$ , and possibly its slope and curvature, at an initial guess  $(0.7067, -3.2672)^T$ , which we call  $x_0$  ( $f(x_0) = 97.6305$ ); we have represented  $x_0$  by a small circle in the figure (near the middle of the bottom line). We now will try to find  $x_*$  by applying a numerical method, an *algorithm*. Based on this information, we may try to guess the shape of the objective function in a neighbourhood of  $x_0$ . In other words, we build around  $x_0$  a *model* of the objective function and decide, admittedly with some degree of arbitrariness, on a region containing  $x_0$  in which we believe the model represents the objective function more or less adequately. This region is called the *trust region* because

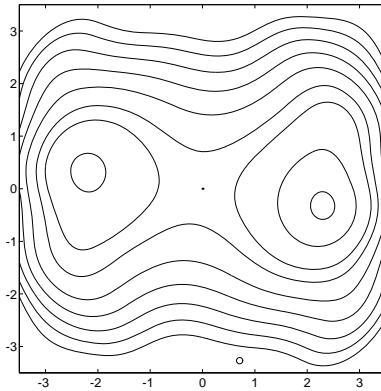


Figure 1.1.1: The contour lines of the objective function, showing two local minima.

this is where we trust the model to be a faithful representation of the objective function. The contour lines of this model are shown in Figure 1.1.2, inside a circle centered at  $x_0$ , which represents the trust region.

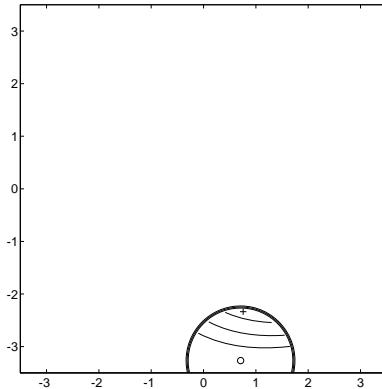


Figure 1.1.2: The model and trust region around  $x_0$ .

Examining the model within this trust region, we notice that its value appears to decrease if we increase the value of  $[x]_2$ , that is, if we move upwards in the figure. We may even select a *step* towards the *trial point*, indicated in Figure 1.1.2 by a small +, where the *decrease in the model* (compared to its value at  $x_0$ ) is significant. At this stage, we have done all we could possibly do with the information at our disposal, and are thus led to compute the objective function value at this new point. As it turns out, the decrease predicted by the model has been partly achieved for the objective function, and the value of the objective function at the new trial point is 43.7420, which is lower than  $f(x_0)$ . We thus decide to move to this new point  $(0.7565, -2.3361)^T$ , call it  $x_1$ , and represent it by a small circle, as it has become our best approximate minimizer so far. Again using the slope and curvature of  $f$  at  $x_1$ , we also construct a new updated model of the objective function around this point. Moreover, considering our success,

we decide to be a little more daring and *increase the radius of the new trust region* centered at  $x_1$ , hoping that our model still represents the objective function well in this enlarged region. The contour lines of the new model within the enlarged trust region are now shown in Figure 1.1.3.

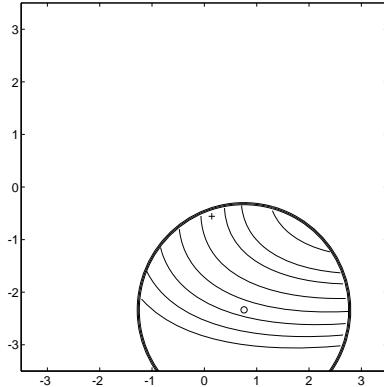


Figure 1.1.3: The model and trust region around  $x_1$ .

As for  $x_0$ , we now examine the model within the trust region and decide that the trial point again indicated by a small + in Figure 1.1.3 provides a sufficient model reduction to justify evaluating the true objective function at this point. As before, this new value (2.3060) is acceptable because it produces a reasonable reduction in the objective function (compared to  $f(x_1)$ ). We thus move to this new point  $(0.1413, -0.5570)^T$ , which we call  $x_2$ , and build a model of the objective function around it using the value  $f(x_2)$ , but also the slope and curvature of  $f$  at  $x_2$ . Capitalizing once more on our success, we further increase the trust-region radius. The result of these operations is illustrated in Figure 1.1.4.

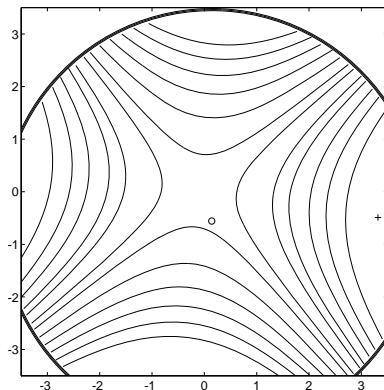


Figure 1.1.4: The model and trust region around  $x_2$ .

We now see that the model has a saddle point within the trust region: while the model increases if we move northwards or southwards<sup>1</sup> (at least with a step that is not

---

<sup>1</sup>That is, changing  $[x]_2$  while keeping  $[x]_1$  fixed.

too small), it decreases if we move eastwards or westwards.<sup>2</sup> Thus it seems reasonable to choose our new trial point (the small +) next to the right limit of the trust region. Unfortunately, this move is too bold, as can be checked in Figure 1.1.1, because the model no longer represents the objective function well this far from  $x_2$  and because the objective function value goes up. Moving to the trial point may therefore not be such a good idea, and we thus stay where we are ( $x_3 = x_2$ ). Reducing our ambition, we then prefer to be slightly more conservative and settle for a shorter step. We thus reduce the trust-region radius and pick a new trial point next to its updated right border, as shown in Figure 1.1.5.

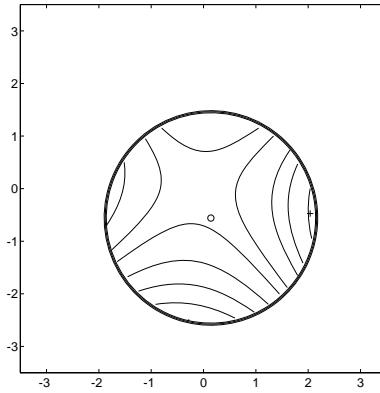


Figure 1.1.5: The model and trust region around  $x_3 = x_2$ .

We are now successful in that the achieved reduction in  $f$  is comparable to the predicted reduction on the model, and the value of the objective function ( $-29.3919$ ) is best at the trial point. We thus call it  $x_4 = (2.0368, -0.4739)^T$  and adopt it as our new current guess. As before, we build a new model around  $x_4$ , based on  $f(x_4)$ , local slope, and curvature, but decide to keep the current trust-region radius. The resulting situation is that of Figure 1.1.6.

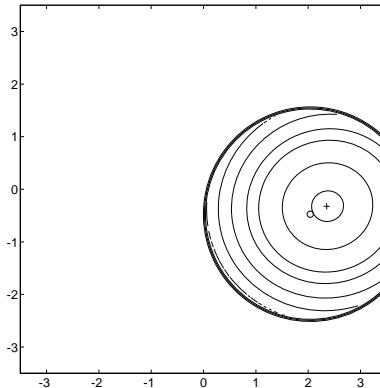


Figure 1.1.6: The model and trust region around  $x_4$ .

---

<sup>2</sup>That is, changing  $[x]_1$  while keeping  $[x]_2$  fixed.

The model now has a clear minimum at  $(2.3195, -0.3409)^T$  within the trust region and it is most natural to choose it as our trial point, as shown in Figure 1.1.6. Again, this move is successful because the model predicts the objective function ( $-31.1315$ ) well at the trial point. We thus move to the new point, now called  $x_5$ , again build a local model around it, and increase the trust-region radius. As illustrated by Figure 1.1.7, the minimum of the model is now very close to  $x_5$ .

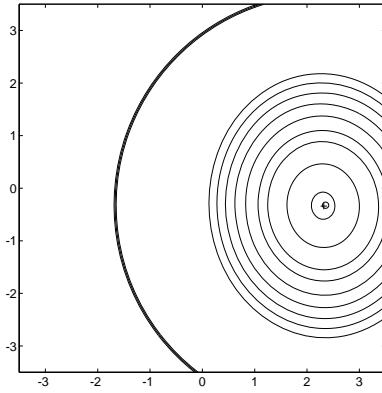


Figure 1.1.7: The model and trust region around  $x_5$ .

A final move towards this model minimizer is now again successful, and a last step of our procedure, illustrated by Figure 1.1.8, enables us to find the rightmost minimizer of the objective function, that is,  $x_7 = x_* = (2.3029, -0.3409)^T$  and  $f(x_*) = -31.1791$ .

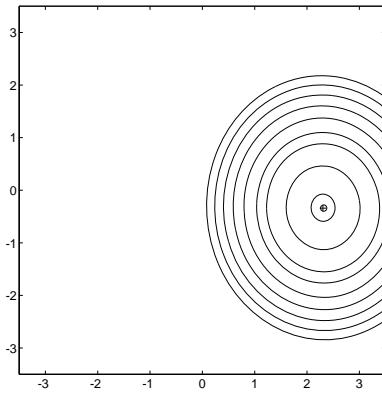


Figure 1.1.8: The model and trust region around  $x_6$ .

Notice that the trust-region radius has now been increased to the point where it completely disappears from the region pictured in the figures. We have used eight evaluations of the objective function (together with its slope and curvature) to reach one of the two local minimizers visible in Figure 1.1.1.

We may now summarize the main features of the procedure we have used in this example—the features that make it a trust-region method.

- The method is *iterative* in that it builds better and better guesses, or *iterates*,  $x_k$  of the solution of the problem.
- It produces a *local solution*. Indeed, we have found in our example one of the two local minimizers of Figure 1.1.1, but nothing guarantees that the objective function does not actually have another, better minimizer outside the area represented in the picture. Moreover, the method that we have followed could also approach the other minimizer if we chose our initial guess and therefore trial points in a slightly different way. This is illustrated by the alternative sequence of iterates shown in Figure 1.1.9.
- At each iterate  $x_k$ , we build a *model that approximates the objective function* in a region centered at  $x_k$ . This region is called the *trust region* and is usually defined as the set of points whose distance to  $x_k$  (in some predefined norm) is at most a given *trust-region radius*. In our example above, we have used the familiar Euclidean norm.
- We then compute a step to a trial point within the trust region at which we obtain a *sufficient model decrease*.
- After computing the value of the objective function at the trial point, we compare the *achieved reduction in  $f$  to the predicted reduction in the model*. If the ratio of achieved versus predicted reduction is sufficiently positive, we define our next guess to be the trial point.
- If this ratio is not sufficiently positive, we decrease the trust-region radius in order to make the trust region smaller. Otherwise, we may increase it or possibly keep it unchanged.

This book is devoted to the study of methods that share these characteristics. They are of considerable interest to the practitioner and theoretician alike because they combine practical efficiency and robustness with strong theoretical support.

As can be suspected, the framework that we have outlined has many variants. We will discover them as we progress through the book, but we may immediately note some of the ways in which they may differ.

- They may be designed for solving different types of problems (unconstrained or constrained optimization, complementarity problems, systems of nonlinear equations, smooth or nonsmooth, etc.).
- The norm defining the trust-region shape may vary in order to exploit the geometry of the underlying problem as effectively as possible.
- The way in which the model is built may be different, according to the kind of available information on the objective function.

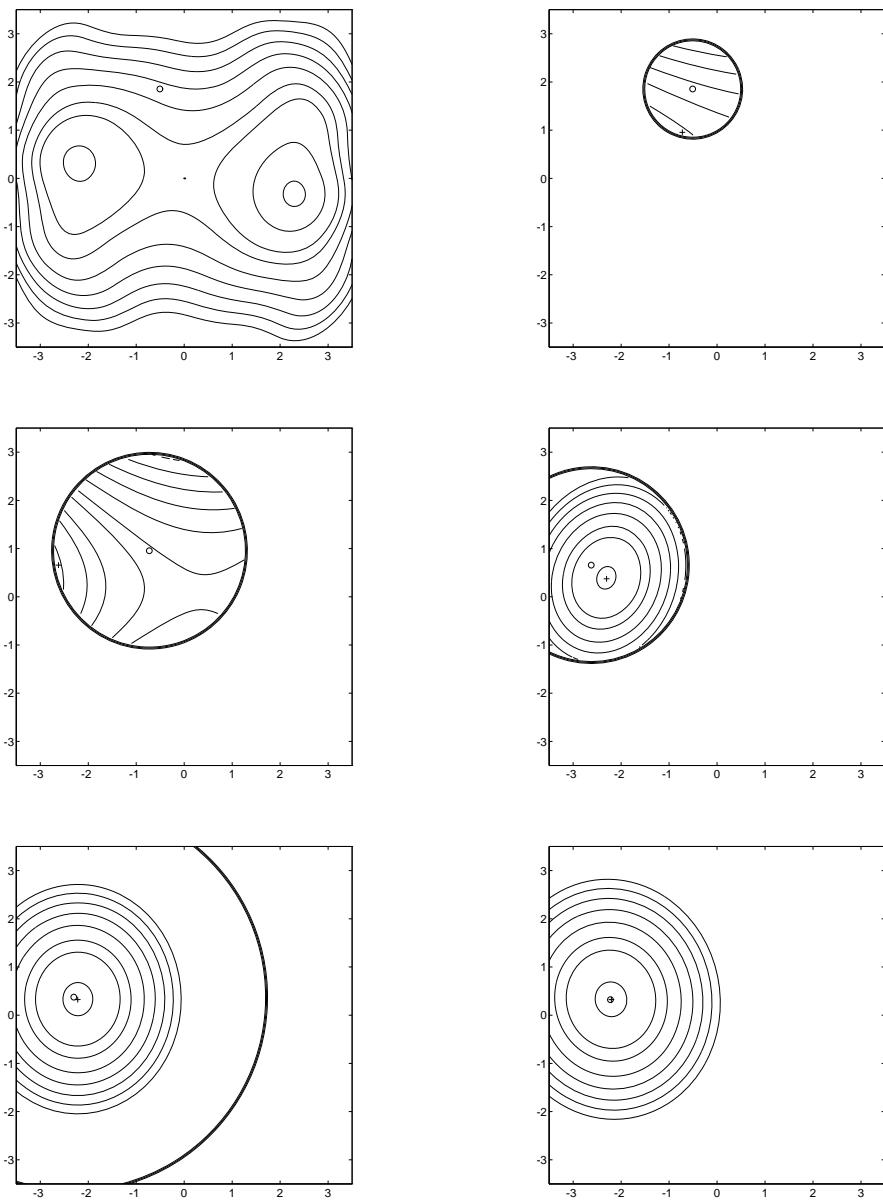


Figure 1.1.9: The six iterations of an alternative execution of the trust-region algorithm (from left to right and top to bottom), where the trust-region radius is increased at every iteration.

- The technique used for computing the trial points at each iteration may also vary, depending on the size or structure of the problem.

This list is far from being complete, as we will discover later, but we hope the reader has already begun to suspect the broad nature and wide applicability of the idea.

## 1.2 Historical Development of Trust-Region Methods

The concept of the trust region has matured over 50 years and is rooted in the field of nonlinear parameter estimation. Early formulations were sometimes quite different from the modern form used in our presentation. The first paper in this area appears to be that of Levenberg (1944), who considers adding a multiple of the identity to the Hessian matrix as a stabilization/damping procedure in the context of the solution of nonlinear least-squares problems. This point of view was developed further (and seemingly independently) by Morrison (1960) in a seldom-mentioned paper on trajectory tracking, in which convergence of the estimation algorithm is enhanced by minimizing a quadratic model of the least-squares objective function in a sphere of constant radius. Morrison proved that the solution of a linear system involving the Hessian of the model augmented by a multiple of the identity matrix yields the solution of the resulting subproblem for a suitable choice of the associated damping parameter, and also that the predicted model decrease is monotone in this parameter. A technique based on eigenvalue decomposition is given to compute the model's minimum within the chosen sphere. The same fundamental idea is found in the celebrated<sup>3</sup> paper by Marquardt (1963), who, again independently, pointed out the link between damping the Hessian and reducing the length of the step and also proved that minimizing a damped model corresponds to minimizing the original model in a restricted region. All three papers make the observation that the Hessian may be approximated, which then yields a quadratic model of the objective function. Another important conceptual contribution is that of Griffith and Stewart (1961), who introduced, apparently unaware of the previous papers, the idea of successively minimizing models of the objective function (and constraints) in a region around the current iterate. Their model is linear and the region is defined by bounds on the variables, leading to a linear programming subproblem. Their paper contains very clear pictures that are close in spirit to Figures 1.1.1–1.1.9. However, they do not propose adjusting the trust-region size, but instead advocate keeping it constant throughout the calculation.

A crucial step was made by Goldfeldt, Quandt, and Trotter (1966), who introduced an explicit updating procedure for the maximum stepsize. Whether they viewed this parameter as Hessian damping, which induces a stepsize restriction, or as a stepsize restriction computed by damping the Hessian is not entirely clear from the paper. However, the updating procedure is indeed very similar to that which we currently use. In particular, they introduce the notion of “achieved versus predicted change”, a quantity that we have used in the example above. There is no doubt that *their contribution really set the stage for trust-region methods in the form that we know today*. Quite remarkably, their paper does not quote any of the previous contributions of Levenberg, Morrison, Marquardt, or Griffith and Stewart. The trust-region concept

---

<sup>3</sup>According to Garfield (1990), cited by Higham (1993, p. 91), this is the 92nd most cited paper in the *Science Citation Index* 1945–1988.

was subsequently advocated by Powell (1970d) as a tool to ensure convergence of a method for unconstrained optimization, where the Powell-symmetric-Broyden (PSB) quasi-Newton formula was used to construct the Hessian of a quadratic model. Powell refers to Marquardt as the inspiration for his proposal. The unpublished paper by Fletcher (1970b) follows the same lines. Powell (1970c) also proposed using a similar device in the context of solving nonlinear equations using the Broyden quasi-Newton update. In these contributions, the updating procedure is clearly focussed on the trust-region radius. One should also note the independent proposal of Winfield (1969, 1973), whose Harvard thesis of 1965–1969 considered the use of a simple but clear trust-region mechanism using a quadratic model in the context of function minimization by interpolation in a data table. From 1970 onwards, the subject was taken on actively by the research community, resulting in a proliferation of results and ideas.

The name *trust region* was not mentioned by Powell (1970b) in his seminal paper, nor by any of his predecessors. The terms *trust region* and *Cauchy point*<sup>4</sup> seem to have been coined by Dennis, in a course he taught at the U.S. National Bureau of Economic Research (NBER) shortly after he heard Powell talk about his technique for solving nonlinear equations. It seems that the first official appearance of the terms *trust region* and *Cauchy point* was in Dennis (1978). However, this terminology did not diffuse to the research community for several years. For instance, Winfield (1973) speaks about a *region of validity* for a quadratic model, Fletcher (1980) uses the term *restricted step method* to indicate that the step is restricted to belonging to a certain region centered at the current iterate, and Toint (1981b) refers to a *confidence region*. The excellent survey of Moré (1983) was influential in standardizing the terminology.

The paper of Powell (1970d) proved global convergence for his trust-region algorithm. Further early theoretical and algorithmic papers include Hebden (1973), Powell (1975), Madsen (1975), Osborne (1976), Moré (1978), Toint (1978), Dennis and Mei (1979), Toint (1979, 1980), parts of Sections 4.4 and 4.7.3 in Gill, Murray, and Wright (1981), Moré and Sorensen (1983), Toint (1981a, 1981b), Sorensen (1982a, 1982b), part of Chapter 6 in Dennis and Schnabel (1983), Moré (1983), Steihaug (1983), and Powell (1984).

### 1.3 Some Applications of Trust-Region Algorithms

While the concept and understanding of trust-region methods was progressing, the idea was also applied in a steadily widening range of problems. It would be futile to even try to cover all applications of trust-region methods in the many fields of science and engineering, not to mention medicine, psychology, and other domains. It is equally impossible to review all applications of trust-region-based packages (see Section 17.5), although they all contribute to the ubiquitous presence of trust-region technology in scientific computations. However, we present in the tables below some references to selected applications of trust-region algorithms, in order to emphasize their diversity and multiplicity.

---

<sup>4</sup>Which we will encounter for the first of many times in Section 6.3.

Applied mathematics	
General optimization	
0-1 programming	Mukai and Polak (1975), Karisch, Rendl, and Wolkowicz (1994), Poljack, Rendl, and Wolkowicz (1995), Poljack and Wolkowicz (1995), Mauricio and Maculan (1997), Mukai, Tatsumi, and Fukushima (1998)
Min-cost flows	Frangioni and Gallo (1999)
Bi-level programming	Case (1997), Alexandrov (1998), Liu, Han, and Wang (1998), Scholtes and Stöhr (1999)
Least-distance problems	Helfrich and Zwick (1995)
Multidisciplinary	Alexandrov and Dennis (1994a, 1994b), Alexandrov et al. (1998), Conn, Scheinberg, and Toint (1998), Booker et al. (1999)
Numerical analysis	
Boundary value problems	Dean (1992)
Nonlinear parameter estimation	Schleiff (1995), Bock, Schlöder, and Schulz (1995)
Homotopy calculations	Watson, Billups, and Morgan (1987)
Curve and surface fitting	Helfrich and Zwick (1996)
Numerical integration	Butcher, Jackiewicz, and Mittelmann (1997)
Fixed-point methods	Lucia, Guo, and Wang (1993)
Partial and ordinary differential equations	Dennis and Williamson (1988), Smith and Bowers (1993), Edsberg and Wedin (1995), Lukšan and Vlček (1996), Kelley and Keyes (1998)
Statistics	Vandergraft (1985), Weihs, Calzolari, and Panattoni (1987), Ekblom and Madsen (1989), Martínez (1995), Edlund (1997), Edlund, Ekblom, and Madsen (1997), Gao (1998), Andrews and Vicente (1999)
Optimal control and system theory	Fukushima and Yamamoto (1986), Propoi and Pukhlikov (1993), Borggaard (1994), Semple (1997), Coleman and Liao (1995), Liao (1995, 1997), Leibfritz and Sachs (1999)
Physics	
Spectroscopy	Bockmann (1996)
Fluid dynamics	Clermont et al. (1991), Borggaard and Burns (1997b)
Optics	Barakat and Sandler (1992, 1999), Lannes (1997, 1998)
Geophysics, seismology, and hydrology	Roger, Terpolilli, and Gosselin (1988), Zhu and Brown (1987), Vogel (1990), Sebudandi (1992), Sebudandi and Toint (1993), Hanke (1997), Rojas (1998)
Electromagnetism	Kojima (1993)

Chemistry	
Physical chemistry	Jensen and Agren (1986), Helgaker and Almlöf (1988), Sun and Ruedenberg (1993)
Chemical engineering	Cesar et al. (1991), Lucia and Xu (1990, 1994), Lucia, Xu, and Layn (1996), Gopal and Biegler (1997), Gow et al. (1997)
Study of transition states	Helgaker (1991)
Chemical kinetics	Mentel and Anderson (1991), Mentel et al. (1992)
Molecular modelling	Bofill (1995)
Crystallography	McAfee et al. (1986, 1988)
Mass transfer	Ji et al. (1999)
Speciation	Holstad (1999)
Engineering	
Structural engineering	Kwok, Kamat, and Watson (1985), Watson, Kamat, and Reaser (1985), Jonsson and Larsson (1990), Sunar and Belegundu (1991), Stander and Snyman (1993), Coster, Stander, and Snyman (1996), Ben-Tal and Zibulevsky (1997), Rodríguez, Renaud, and Watson (1999)
Transportation analysis	Bierlaire and Toint (1995)
Energy distribution networks	Tappayuthpijarn and Jalali (1990), Furey (1993), Shiina (1999)
Radar applications	Jain, McClellan, and Sarkar (1986)
Modelling and mechanical design	Lewis (1996), Cheng and Sykulski (1996), Petzold, Ren, and Maly (1997), Borggaard and Burns (1997a), Rodríguez, Renaud, and Watson (1998)
Circuit design	Bandler, Chen, and Madsen (1988), Pornbacher et al. (1990), Rudnicki (1994), Conn et al. (1996), Bakr et al. (1998, 1999), Schwenker et al. (1999)
Computer science	
Neural network training	Pham Dinh and Wang (1990), Bulsari, Sillanpää, and Saxen (1992), Kojima and Kawaguchi (1993), Zhao and Wang (1993), Madhyastha and Aazhang (1994), Tsoutsias and Mjolsness (1996), Zhou and Si (1998, 1999)
Computer vision	Phong et al. (1995), Le Thi (1997), Pham Dinh et al. (1997)
Motion estimation and robotics	Jagersand (1995), Mallick (1997), Jagersand, Fuentes, and Nelson (1996, 1997), Piepmeyer, McMurray, and Lipkin (1998)

---

Biology and medicine	
Magnetic resonance applications	Budil et al. (1996)
Rheology	Bereaux and Clermont (1997)
Pharmacokinetics	Allen (1995)
Computer-aided diagnostics	Kehtarnavaz, Win, and Mullani (1987)
Optical tomography	Roy and Sevick Muraca (1999)
Economics and sociology	
Random utility models	Bierlaire (1994, 1995, 1998)
Game theory and international negotiations	Germain and Toint (2000)
Financial portfolio management	Studer and Lüthi (1997), Sachs, Schulze, and Fromme (1998), Studer (1999)

---

We believe that there will be numerous further developments in even more diverse application areas in the coming years.

# **Part I**

---

## **Preliminaries**

In this first part of our book, we review all of the background material upon which later sections will rest. Almost all of the material here is well known, and this part of the book is intended to be self-contained. There is little in the way of proof here, but there is a comprehensive list of relevant references to help the reader in distress. The advanced reader may wish to skip much of the material given here and use it merely as a source of reference.

---

# Chapter 2

---

---

## Basic Concepts

---

---

The purpose of this and the succeeding three chapters is to set the stage for what follows. We use two basic tools, linear algebra and real analysis, throughout the book. Of course, both disciplines are vast, and we have chosen to review only those topics that are directly relevant to the rest of our book. However, even with this goal in mind, the level of exposition we offer will vary from topic to topic. This is deliberate. Some of the material we review is so basic that we feel it inappropriate that we should more than summarize what is known. If the reader feels we are going too fast, we provide references to the books and papers where this material is covered in considerably more depth. The remainder of the material is more directly relevant, and in this case we try to motivate and develop it so that we hope the reader need look no further.

### 2.1 Basic Notation

We shall denote the  $i$ th component of a vector  $x \in \mathbb{R}^n$  by  $[x]_i$ , or simply  $x_i$  if no confusion can arise from other suffixes attached to  $x$ . Similarly, the  $(i, j)$ th component of the matrix  $A \in \mathbb{R}^{m \times n}$  will be written as  $[a]_{i,j}$  or simply  $a_{i,j}$  (note that matrices are traditionally written in upper case, while their components are in lower case). If  $\mathcal{S}$  is a subset of  $\{1, \dots, n\}$ , we shall write  $[x]_{\mathcal{S}}$  or simply  $x_{\mathcal{S}}$  for the vector whose components are  $x_i, i \in \mathcal{S}$ .

There are a number of scalar functions that we shall extend quite naturally for vectors. The vector  $|x|$  is simply that whose  $i$ th component is  $|x_i|$ . Similarly, we shall write  $x^+$  for the vector whose  $i$ th component is  $\max[x_i, 0]$ , and  $x^-$  for that whose  $i$ th component is  $\min[x_i, 0]$ . The vectors  $\max[x, y]$  and  $\min[x, y]$  are simply those whose  $i$ th components are  $\max[x_i, y_i]$  and  $\min[x_i, y_i]$ , respectively, while  $\text{mid}(x, y, z)$  is that whose  $i$ th component is the median of  $x_i, y_i$ , and  $z_i$ . The Kronecker delta  $\delta_{i,j}$  is defined to be

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise} \end{cases}$$

for any pair of integer indices  $i$  and  $j$ . The matrix  $I_n$  will denote the  $n$  by  $n$  *identity* matrix, that is, the  $n$  by  $n$  matrix whose  $(i, j)$ th entry is  $\delta_{i,j}$ . The  $i$ th column of  $I_n$  will be given by  $e_i$  and will sometimes be referred to as the  $i$ th canonical basis vector of  $\mathbb{R}^n$ . When the dimension is clear from the context, we may write  $I$  instead of  $I_n$ .

For  $\alpha \in \mathbb{R}$ , we also define

$$\operatorname{sgn}(\alpha) = \begin{cases} +1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \\ -1 & \text{if } \alpha < 0. \end{cases}$$

We denote by  $\lfloor \alpha \rfloor$  the largest integer that is not larger than  $\alpha$  (i.e.,  $\alpha$  rounded down) and by  $\lceil \alpha \rceil$  the smallest integer that is not smaller than  $\alpha$  (i.e.,  $\alpha$  rounded up).

Although we shall discuss inner products and norms in some detail in Section 2.3, in the interim we shall refer to the Euclidean inner product

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i$$

of two vectors  $x$  and  $y$ , and to the related Euclidean vector norm  $\|x\|_2 \stackrel{\text{def}}{=} \sqrt{\langle x, x \rangle}$  of  $x$ . Throughout the book, unless we say otherwise, the symbol  $\|\cdot\|$  will denote the Euclidean vector norm or its subordinate matrix norm (see Section 2.3.2).

## 2.2 Eigenvalues and Eigenvectors

Given an  $n$  by  $n$  matrix  $A$ , the solutions  $(u, \lambda)$  ( $u \neq 0$ ) to the nonlinear equation

$$Au = \lambda u \tag{2.2.1}$$

are known as *eigenpairs*, the scalar  $\lambda$  is known as an *eigenvalue*, the vector  $u$  is an *eigenvector*, and the problem itself is an *eigenproblem*. As  $\alpha u$  is also an eigenvector for any  $\alpha \neq 0$ , it is common to normalize eigenvectors so that  $\|u\| = 1$ , and when we refer to eigenvectors, we shall implicitly assume that they are normalized.

There are many related definitions of eigenvalues, of which the best known is that the eigenvalues necessarily satisfy the equation  $\det(A - \lambda I) = 0$ . This immediately implies that the eigenvalues satisfy a polynomial equation of degree  $n$ , and thus that  $A$  has  $n$  eigenvalues. Notice, however, that the eigenvalues need not be real, although any complex ones must appear as conjugate pairs. The set of eigenvalues of a matrix is known as its *spectrum*.

If  $R$  is a nonsingular matrix, the matrices  $A$  and  $R^{-1}AR$  have the same eigenvalues, whereas they have eigenvectors  $u$  and  $R^{-1}u$ , respectively. Any two matrices related in this way are said to be *similar*, and  $R^{-1}AR$  is a *similarity transform* of  $A$ .

We shall be concerned only with real matrices. However, even given this restriction, there are significant differences between the eigenpairs of symmetric and unsymmetric matrices. The eigenvalues of real symmetric matrices are necessarily real, and it is

possible to find a complete set of  $n$  orthonormal eigenvectors. Using the superscript  $T$  to denote the matrix transpose, we may thus write

$$H = U\Lambda U^T \quad (2.2.2)$$

for any real symmetric  $H$ , where the entries of the diagonal matrix  $\Lambda$  are the eigenvalues of  $H$  and the columns of the orthonormal matrix<sup>5</sup>  $U$  are the associated eigenvectors—the relationship (2.2.2) is sometimes known as the eigendecomposition or spectral decomposition of  $H$ . Real symmetric matrices with strictly positive eigenvalues are known as *positive definite* matrices and will play a central role in this book. If the eigenvalues of a real, symmetric matrix are nonnegative, it is a *positive semidefinite* matrix. If such a matrix has both positive and negative eigenvalues, we shall say that it is *indefinite*. Note that it follows directly that a real symmetric matrix  $H$  is positive semidefinite if and only if

$$\langle s, Hs \rangle \geq 0$$

for all vectors  $s$ , while the matrix is positive definite if and only if

$$\langle s, Hs \rangle > 0$$

for all nonzero vectors  $s$ . Counting the numbers of positive and negative eigenvalues of a real symmetric matrix will prove to be revealing in later sections of the book. To this end, we can record the *inertia* of  $H$ ,  $\text{In}(H) = (h_+, h_-, h_0)$ , where  $h_+$ ,  $h_-$ , and  $h_0$  are, respectively, the numbers of positive, negative, and zero eigenvalues of  $H$ . Clearly,  $h_+ + h_- + h_0 = n$ . Two other important matrices that may be derived from (2.2.2) will also be useful to us. When  $H$  is symmetric, we say that

$$|H| = U|\Lambda|U^T \quad (2.2.3)$$

is the *absolute value* of  $H$ , where the absolute value  $|\Lambda|$  of the diagonal matrix  $\Lambda$  is simply the matrix formed by taking absolute values of its entries. If, in addition,  $H$  is positive semidefinite, we say that

$$\sqrt{H} \stackrel{\text{def}}{=} H^{\frac{1}{2}} = U\sqrt{\Lambda}U^T$$

is the *square root* of  $H$ , where this time the square root  $\sqrt{\Lambda}$  of  $\Lambda$  is the matrix formed by taking (positive) square roots of its entries.

Unsymmetric matrices, on the other hand, may have complex-conjugate pairs of eigenvalues and, more significantly, may not even have a complete set of eigenvectors. There is, however, a decomposition related to (2.2.2) in the unsymmetric, and even nonsquare, case. The *singular value* decomposition of a real  $m$  by  $n$  matrix  $A$  is a factorization of the form

$$A = U\Sigma V^T, \quad (2.2.4)$$

---

<sup>5</sup>Unfortunately most authors refer to these matrices as orthogonal matrices. As the columns are also required to be normalized, we prefer the more specific epithet given here.

where  $U$  and  $V$  are, respectively,  $m$  by  $m$  and  $n$  by  $n$  orthonormal matrices, and  $\Sigma$  is a real  $m$  by  $n$  diagonal matrix with nonnegative entries; the columns of  $U$  and  $V$  are known as the left and right *singular vectors* of  $A$ , while the diagonal entries are its *singular values*. The singular values/vectors of  $A$  are related to the eigenpairs of  $AA^T$  and  $A^TA$ . Specifically, the squares of the nonzero singular values of  $A$  are the eigenvalues of  $AA^T$  (and  $A^TA$ ), while the left and right singular vectors are the eigenvectors, respectively, of  $AA^T$  and  $A^TA$ . This can be seen immediately from the relationships

$$AA^T = U(\Sigma\Sigma^T)U^T \quad \text{and} \quad A^TA = V(\Sigma^T\Sigma)V^T,$$

which follow from the definition (2.2.4). The singular value decomposition reveals the *rank* of a matrix  $A$ ,  $\text{rank}(A)$ , which is simply the number of nonzero singular values of  $A$ —it is also the maximum number of linearly independent<sup>6</sup> rows (and columns) of  $A$ . The matrix is said to be of *full rank* if  $\text{rank}(A) = \min[m, n]$ , and is otherwise *rank deficient*.

As the defining relationships (2.2.2) and (2.2.4) are nonlinear, it is too much, in general, to hope that there will be finite methods to compute these decompositions. There are, however, many effective methods that compute approximate eigen- and singular values and their associated vectors, and these frequently suffice for the applications we shall encounter.

Throughout this book we shall denote the  $i$ th leftmost eigenvalue of the real, symmetric matrix  $H$  as  $\lambda_i[H]$ , or  $\lambda_i$  when it is clear from the context which real symmetric matrix is involved. Its associated eigenvector will be  $u_i[H]$ , and the *determinant* of  $H$ ,

$$\det(H) = \prod_{i=1}^n \lambda_i,$$

is defined to be the product of all its eigenvalues. The left- and rightmost eigenvalues of  $H$  will be specially designated  $\lambda_{\min}[H]$  and  $\lambda_{\max}[H]$ , or simply  $\lambda_{\min}$  and  $\lambda_{\max}$  when the meaning is clear. Similarly, we shall label the  $i$ th smallest singular value of the real matrix  $A$  as  $\sigma_i[A]$ , or as  $\sigma_i$  when the context is clear. The smallest and largest singular values of  $A$  will be written as  $\sigma_{\min}[A]$  and  $\sigma_{\max}[A]$ , or sometimes simply  $\sigma_{\min}$  and  $\sigma_{\max}$ .

There is a most important relationship between the eigenvalues of a symmetric matrix and those of its symmetric subblocks. Suppose that  $H$  is a symmetric  $n$  by  $n$  matrix and that

$$K = \begin{pmatrix} H & h \\ h^T & \theta \end{pmatrix},$$

where  $h$  is any vector and  $\theta$  any scalar. Then the eigenvalues of  $H$  and  $K$  satisfy the inequalities

$$\lambda_1[K] \leq \lambda_1[H] \leq \lambda_2[K] \leq \cdots \leq \lambda_n[K] \leq \lambda_n[H] \leq \lambda_{n+1}[K].$$

---

<sup>6</sup>A set of vectors  $\{v_i\}$ ,  $i = 1, \dots, k$ , are *linearly independent* if and only if the only set of scalars  $\{\alpha_i\}$  for which  $\sum_{i=1}^k \alpha_i v_i = 0$  are the values  $\alpha_i = 0$ . If there is a nonzero set of  $\{\alpha_i\}$ , the vectors are *linearly dependent*.

This is known as the Cauchy interlacing property, or simply the *interlacing* property, for bordered matrices.

If  $H$  is symmetric and the vector  $p \neq 0$ , the scalar

$$\frac{\langle p, Hp \rangle}{\langle p, p \rangle}$$

is known as the *Rayleigh quotient* of  $p$ . The Rayleigh quotient is important because it lies between the left- and rightmost eigenvalues of  $H$ , that is,

$$\lambda_{\min}[H] \leq \frac{\langle p, Hp \rangle}{\langle p, p \rangle} \leq \lambda_{\max}[H] \quad (2.2.5)$$

for any nonzero  $p$ . Inequality (2.2.5) is known as the Rayleigh quotient inequality; the minimum and maximum values are attained by the eigenvectors corresponding to the left- and rightmost eigenvalues, respectively.

While, as we have suggested, it may be expensive to obtain the eigenvalues of a matrix, it is often straightforward to obtain useful bounds on them. Of the large number of proposed bounds, the best known are the *Gershgorin* bounds. These state that all the eigenvalues of a matrix  $A$  lie in the complex plane within the intersection of  $n$  discs centered at  $a_{i,i}$  and of radii  $\sum_{j \neq i} |a_{i,j}|$  for  $1 \leq i \leq n$ . If a disc is isolated, it contains precisely one eigenvalue. When the matrix  $H$  is symmetric (and thus the eigenvalues are real), the discs become intervals on the real line and provide the crude but sometimes useful and easily computable bounds

$$\min_i \left( h_{i,i} - \sum_{j \neq i} |h_{i,j}| \right) \leq \lambda_{\min}[H] \leq \lambda_{\max}[H] \leq \max_i \left( h_{i,i} + \sum_{j \neq i} |h_{i,j}| \right). \quad (2.2.6)$$

The problem (2.2.1) is a special case of the *generalized* eigenproblem

$$Au = \lambda Bu, \quad (2.2.7)$$

where  $B$  is a second  $n$  by  $n$  matrix; a solution  $(u, \lambda)$  with  $u \neq 0$  is a *generalized* eigenpair of  $(A, B)$ , while its components are a *generalized* eigenvector and eigenvalue, respectively. Determining the existence of nonzero solutions to (2.2.7) is often hard. In the special case where  $H$  is symmetric and  $M$  symmetric positive definite, however, it is straightforward in principle to reduce the generalized problem  $(H, M)$  to an ordinary symmetric problem, and thus to ensure that there are  $n$  real (generalized) eigenvalues. In this case, we shall denote the  $i$ th leftmost generalized eigenvalue and eigenvector of the pair  $(H, M)$  as  $\lambda_i[H, M]$  and  $u_i[H, M]$ , respectively.

Finally, while it may not be efficient to do so, the spectral decomposition (2.2.2) and its unsymmetric analog, the singular value decomposition (2.2.4), may in principle be used to solve linear systems of equations. For example, if  $A$  is nonsingular and has a singular value decomposition of the form (2.2.4), the solution to the linear system  $Ax = b$  is given (formally) by

$$x = V\Sigma^{-1}U^T b.$$

Of perhaps more use is the fact that if  $A$  is singular (or rectangular) but the system consistent, a solution is given by

$$x = V\Sigma^+U^T b, \quad (2.2.8)$$

where the entries of the diagonal  $n$  by  $m$  matrix  $\Sigma^+$  are given by

$$\sigma_{ii}^+ = \begin{cases} 1/\sigma_{ii} & \text{if } \sigma_{ii} \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix

$$A^+ \stackrel{\text{def}}{=} V\Sigma^+U^T$$

is known as the *Moore–Penrose generalized inverse* of  $A$ . We shall see that such matrices play an important role in the solution of trust-region problems. We note that if  $A$  is singular or rectangular and  $Ax = b$  is inconsistent, the value of  $x$  in (2.2.8) gives the *minimum-norm* least-squares solution. That is, it gives the value of  $x$  that minimizes  $\|Ax - b\|_2$  and for which  $\|x\|_2$  is smallest. We shall consider such problems in more detail in Section 4.4.1.

## Notes and References for Section 2.2

The theory and solution of eigenproblems has a vast literature. The fundamental aspects of the symmetric problem are covered by Parlett (1980), while Wilkinson (1965) is the classic reference book on general eigenproblems. Other relevant references include the books by Golub and Van Loan (1989) and Saad (1991).

Numerical methods for computing eigenpairs and the singular value decomposition for small matrices have been implemented as part of the EISPACK (Smith et al., 1976), LINPACK (Dongarra et al., 1979), and LAPACK (Anderson et al., 1995) software packages. The principal tool for larger matrices, the Lanczos (1950) method, will be discussed in Section 5.2.

## 2.3 Vector and Matrix Norms

We use norms to measure the size of vectors, matrices, and functions. A norm is a function  $\|\cdot\| : \mathcal{S} \rightarrow \mathbb{R}$  which is required to satisfy three basic properties:

- (a)  $\|x\| \geq 0$  and  $\|x\| = 0 \iff x = 0$ ;
- (b)  $\|x + y\| \leq \|x\| + \|y\|$ ;
- (c)  $\|\alpha x\| = |\alpha| \|x\|$

for all  $\alpha \in \mathbb{R}$  and  $x$  and  $y$  in the relevant space  $\mathcal{S}$ . Property (b) is known as the triangle inequality.<sup>7</sup>

---

<sup>7</sup>More generally,  $\alpha$  could be allowed to be complex, but this is unnecessary for our purposes.

### 2.3.1 Vector and Functional Norms

When  $\mathcal{S}$  is  $\mathbb{R}^n$ , the most common norms are the  $\ell_p$  vector norms<sup>8</sup> ( $p \geq 1$ ), defined by

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

The most common of these are the vector  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms

$$\|x\|_1 = \sum_{i=1}^n |x_i|, \quad \|x\|_2 = \sqrt{\langle x, x \rangle}, \quad \text{and} \quad \|x\|_\infty = \max_{1 \leq i \leq n} |x_i|,$$

where  $\langle x, y \rangle$  denotes the *inner product*

$$\langle x, y \rangle = \sum_{i=1}^n x_i y_i$$

on  $\mathbb{R}^n$ . The  $\ell_2$  norm is often known as the *Euclidean* norm.

The  $\ell_p$  and  $\ell_q$  norms are said to be *dual* if  $1/p + 1/q = 1$ . The norms  $\|\cdot\|_1$  and  $\|\cdot\|_\infty$  are dual, while  $\|\cdot\|_2$  is self-dual. The importance of dual norms is that the inner product  $\langle x, y \rangle$  satisfies the Hölder inequality

$$-\|x\|_p \|y\|_q \leq \langle x, y \rangle \leq \|x\|_p \|y\|_q$$

when  $\|\cdot\|_p$  and  $\|\cdot\|_q$  are dual. Important special cases are

$$-\|x\|_1 \|y\|_\infty \leq \langle x, y \rangle \leq \|x\|_1 \|y\|_\infty$$

and the Cauchy–Schwarz inequality

$$-\|x\|_2 \|y\|_2 \leq \langle x, y \rangle \leq \|x\|_2 \|y\|_2.$$

More generally, the dual norm of an arbitrary norm  $\|\cdot\|$  is defined by

$$\|x\|_{\text{D}} = \sup_{\|y\| \leq 1} |\langle y, x \rangle| \equiv \sup_{y \neq 0} \frac{|\langle y, x \rangle|}{\|y\|}, \quad (2.3.1)$$

and the dual of this dual norm is, reassuringly, the original norm. The Hölder inequality for arbitrary norms is then

$$-\|x\| \|y\|_{\text{D}} \leq \langle x, y \rangle \leq \|x\| \|y\|_{\text{D}}.$$

When  $\mathcal{S}$  is  $\mathbb{R}^n$ , all norms are equivalent in the sense that it can be shown that if  $\|\cdot\|$  and  $\|\cdot\|_{\text{N}}$  are any pair of norms, there are constants  $\kappa_l$  and  $\kappa_u$  such that

$$\kappa_l \|x\|_{\text{N}} \leq \|x\| \leq \kappa_u \|x\|_{\text{N}}$$

for all  $x \in \mathbb{R}^n$ . For instance, the  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms satisfy the relationships

$$\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1 \leq \sqrt{n} \|x\|_2 \leq n \|x\|_\infty$$

---

<sup>8</sup>These are often simply called  $p$  norms or Hölder norms.

over  $\mathbb{R}^n$ . The implication here is that it matters little from a geometric viewpoint which norm is used in  $\mathbb{R}^n$ , and thus a norm should be chosen purely for its convenience. Note that norms over infinite-dimensional spaces need not be equivalent.

One further vector norm, the  $H$  norm, is commonly used. Given a positive definite matrix  $H$ , the  $H$  norm<sup>9</sup> of a vector  $x$  is defined to be

$$\|x\|_H = \sqrt{\langle x, Hx \rangle}. \quad (2.3.2)$$

The  $H$  norm may simply be viewed as the Euclidean norm following a change of variables, and it is often used precisely in this context. Figure 2.3.1 illustrates the shape of the unit ball and of the corresponding dual unit ball for an ellipsoidal  $H$  norm in the plane.

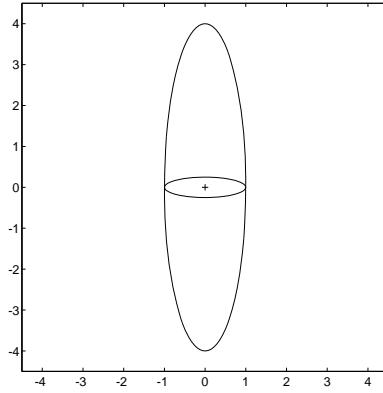


Figure 2.3.1: The shape of the unit ball for an  $H$  norm and its dual in the plane.

### 2.3.2 Matrix Norms

When  $\mathcal{S} = \mathbb{R}^{m \times n}$ , there are two common ways of defining norms. The first is to merely view  $\mathbb{R}^{m \times n}$  as  $\mathbb{R}^{mn}$ , and to use an appropriate vector norm defined on  $\mathbb{R}^{mn}$ . The most frequently used of these is the *Frobenius* or Euclidean matrix norm

$$\|A\|_{\text{F}} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{i,j}^2},$$

while other possibilities include the matrix max and sum norms

$$\|A\|_{\text{M}} = \max_{i,j} |a_{i,j}| \quad \text{and} \quad \|A\|_{\text{S}} = \sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|,$$

---

<sup>9</sup>The term  $H$  is used generically here. If, for instance, the norm is defined in terms of a positive definite matrix  $B$ , say, the norm will be called the  $B$  norm.

where  $A \in \mathbb{R}^{m \times n}$ . The second common way of defining a matrix norm is to consider a vector norm  $\|\cdot\|_v$  and to make the matrix norm *subordinate* to, or *induced* by, its vector counterpart via

$$\|A\|_v = \max_{x \neq 0} \frac{\|Ax\|_v}{\|x\|_v}.$$

The three most frequently used subordinate matrix norms are the matrix  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms, which have the simplified forms

$$\|A\|_1 = \max_{1 \leq i \leq n} \|Ae_i\|_1, \quad \|A\|_2 = \sigma_{\max}[A], \quad \text{and} \quad \|A\|_\infty = \max_{1 \leq i \leq m} \|A^T e_i\|_1,$$

where  $A \in \mathbb{R}^{m \times n}$ . Thus the matrix  $\ell_1$  and  $\ell_\infty$  norms are merely the maximum  $\ell_1$  norms of the columns and rows of  $A$ , while the matrix  $\ell_2$  norm is its largest singular value. Thus we see that while the  $\ell_1$  and  $\ell_\infty$  norms are relatively inexpensive to compute, the  $\ell_2$  norm may be less so. Once again, there are simple relationships

$$\|A\|_\infty \leq \sqrt{n}\|A\|_2 \leq n\|A\|_1 \leq n\sqrt{m}\|A\|_2 \leq nm\|A\|_\infty \quad (2.3.3)$$

between these norms, while in addition, there is now a relationship

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2 \quad (2.3.4)$$

between the Frobenius and matrix  $\ell_2$  norms.

Finally, a matrix norm  $\|\cdot\|$  is said to be *consistent* if

$$\|AB\| \leq \|A\| \|B\| \quad (2.3.5)$$

for all  $A$  and  $B$  for which the product  $AB$  is defined. The Frobenius, sum, and all subordinate norms are consistent, while the max norm is not. The inequality (2.3.5) is often known as the *submultiplicative* property for matrix norms. Another useful inequality

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$$

is often used to provide a further easily computable bound on the Euclidean matrix norm. In particular, this inequality shows that

$$\|H\|_2 \leq \|H\|_1 \equiv \|H\|_\infty \quad (2.3.6)$$

whenever  $H$  is symmetric, which is an improvement on (2.3.3) in this case.

## Notes and References for Section 2.3

Higham (1996, Chapter 6) and Lancaster and Tismenetsky (1985, Chapter 10) give good descriptions of norms on finite-dimensional spaces, as do many other basic numerical analysis textbooks. Collatz (1966) and Cryer (1982) consider both the finite- and infinite-dimensional cases and highlight the subtleties associated with the latter.

# Chapter 3

---

## Basic Analysis and Optimality Conditions

---

In our second chapter of introductory material, we concentrate on one of the two fundamental pillars of optimization, namely real and functional analysis.

### 3.1 Basic Real Analysis

In this section, we briefly remind the reader of those concepts from real analysis and topology which underpin optimization theory.

#### 3.1.1 Basic Topology

Suppose  $\|\cdot\|$  is a norm on  $\mathbb{R}^n$ . We consider a subset  $\mathcal{S}$  of  $\mathbb{R}^n$ . We define an *open ball* of radius  $\epsilon$  about  $x \in \mathcal{S}$  to be the set

$$\mathcal{O}_\epsilon(x) = \{y \mid \|y - x\| < \epsilon\}.$$

The set  $\mathcal{S}$  is said to be *open* in  $\mathbb{R}^n$  if for every  $x$  in  $\mathcal{S}$ , there is a scalar  $\epsilon(x) > 0$  such that  $\mathcal{O}_{\epsilon(x)}(x) \subseteq \mathcal{S}$ . Both the empty set and  $\mathbb{R}^n$  are open in  $\mathbb{R}^n$ , as are open balls and the Cartesian products of open intervals  $(a_i, b_i)$ . The intersection of a finite number of open sets, and the union of an arbitrary number of open sets, is open.

We may also consider closed sets. A set  $\mathcal{S}$  is *closed* in  $\mathbb{R}^n$  if  $\mathbb{R}^n \setminus \mathcal{S}$  is open. Thus, once again, both the empty set and  $\mathbb{R}^n$  are closed in  $\mathbb{R}^n$ . The *closed ball* of radius  $\epsilon$  about  $x \in \mathcal{S}$ ,

$$\mathcal{B}_\epsilon(x) = \{y \mid \|y - x\| \leq \epsilon\},$$

is closed, as are the Cartesian products of closed intervals  $[a_i, b_i]$ . The union of a finite number of closed sets, and the intersection of an arbitrary number of closed sets, is closed.

A set  $\mathcal{S}$  is *bounded* if there is a constant  $\kappa$  for which

$$\|x - y\| \leq \kappa$$

for all  $x, y \in \mathcal{S}$ . A set  $\mathcal{S} \in \mathbb{R}^n$  which is both closed and bounded is said to be *compact*.<sup>10</sup> The main benefits of compactness are that sequences in compact spaces have convergent subsequences and that continuous functions attain their infima and suprema on compact sets.

As  $\mathcal{S}$  may be neither open nor closed, we may be interested in the smallest closed set that contains  $\mathcal{S}$  and the largest open set that is contained in  $\mathcal{S}$ . The first of these is easy to define. The *closure* of  $\mathcal{S}$  is the set of points obtained by extending  $\mathcal{S}$  to its boundary, or more formally,

$$\text{cl}\{\mathcal{S}\} = \{x \mid \text{for all } \epsilon > 0 \text{ } \mathcal{S} \cap \mathcal{B}_\epsilon(x) \neq \emptyset\}.$$

Likewise, we can define the *interior* of  $\mathcal{S}$  as the set of points whose neighbours are also in  $\mathcal{S}$ , or, more formally, the set

$$\{x \in \mathcal{S} \mid \text{there is an } \epsilon > 0 \text{ so that } \mathcal{O}_\epsilon(x) \subseteq \mathcal{S}\}.$$

The problem with this definition of the interior of a set is that it takes no account of the fact that  $\mathcal{S}$  may be of lower dimension than is appropriate for the norm used. As an example, if  $\mathcal{S} = \{x \in \mathbb{R}^2 \mid x \geq 0 \text{ and } x_1 + x_2 = 1\}$ , the interior of  $\mathcal{S}$  is null as there are always points “just off” the line  $x_1 + x_2 = 1$  which are not in  $\mathcal{S}$ . However, if we had restricted our attention to the points that lie on this line, it would be reasonable to say that  $\mathcal{S}$  has an interior, the set  $\{x \in \mathbb{R}^2 \mid x > 0 \text{ and } x_1 + x_2 = 1\}$ . This is what we call the relative interior of  $\mathcal{S}$ .

To state this formally, we need to define the affine hull of a set. An *affine set* is a subset  $\mathcal{A}$  of  $\mathbb{R}^n$  for which  $x + \theta(y - x) \in \mathcal{A}$  for all  $x$  and  $y \in \mathcal{A}$  and any scalar  $\theta$ . Hyperplanes (that is, the set of points

$$\{x \mid \langle a, x \rangle = b\}$$

for some given vector  $a$  and scalar  $b$ ), lines, and single points are affine sets. The *affine hull* of a set  $\mathcal{S}$ ,  $\text{aff}\{\mathcal{S}\}$ , is the intersection of all affine sets that contain  $\mathcal{S}$ . The *relative interior* of  $\mathcal{S}$  is the set of points whose neighbours in the affine hull of  $\mathcal{S}$  are also in  $\mathcal{S}$ , or formally,

$$\text{ri}\{\mathcal{S}\} = \{x \in \mathcal{S} \mid \text{there is an } \epsilon > 0 \text{ so that if } y \in \mathcal{O}_\epsilon(x) \cap \text{aff}\{\mathcal{S}\} \text{ then } y \in \mathcal{S}\}.$$

We then have that

$$\text{ri}\{\mathcal{S}\} \subseteq \mathcal{S} \subseteq \text{cl}\{\mathcal{S}\},$$

and we may define the *relative boundary* of  $\mathcal{S}$  as  $\partial\mathcal{S} \stackrel{\text{def}}{=} \text{cl}\{\mathcal{S}\} \setminus \text{ri}\{\mathcal{S}\}$ . We note that, while

$$\text{cl}\{\mathcal{S}_1\} \subseteq \text{cl}\{\mathcal{S}_2\}$$

is always true when  $\mathcal{S}_1 \subseteq \mathcal{S}_2$ , it does not follow that  $\text{ri}\{\mathcal{S}_1\} \subseteq \text{ri}\{\mathcal{S}_2\}$ . For example, if  $\mathcal{S}_1$  is one of the sides of a square,  $\mathcal{S}_2$ , in  $\mathbb{R}^2$ ,  $\text{ri}\{\mathcal{S}_1\} \not\subseteq \text{ri}\{\mathcal{S}_2\}$ .

<sup>10</sup>In infinite-dimensional spaces, there are more general definitions of compactness. For instance, a set  $\mathcal{S}$  is compact if any arbitrary sequence  $\{x_k\}$  in  $\mathcal{S}$  has a convergent subsequence whose limit is in  $\mathcal{S}$ . There are examples that are closed and bounded but not compact in infinite-dimensional spaces, but the two are synonyms in  $\mathbb{R}^n$ —this is the Heine–Borel theorem.

## Notes and References for Subsection 3.1.1

See Sutherland (1975) for a good introduction to the required elements of topology. More advanced treatments are given by Luenberger (1969), Rockafellar (1970), and Cryer (1982).

### 3.1.2 Derivatives and Taylor's Theorem

Derivatives are the most useful attributes a function can be blessed with when it comes to optimization. Problems without derivatives are almost always harder to solve than their differentiable counterparts. It is safe to say that the more derivatives that are available, the easier it is to solve the problem.<sup>11</sup>

Let  $f(x)$  be a function<sup>12</sup> from  $\mathbb{R}^n$  to  $\mathbb{R}$ . We say that  $f$  is in  $C^k$  if it has continuous  $k$ th-order partial derivatives. If  $f$  is continuously differentiable ( $f \in C^1$ ), we define the *gradient* of  $f(x)$  to be the vector-valued function  $\nabla_x f(x)$ , whose  $i$ th component is  $\partial f(x)/\partial x_i$ . The suffix  $x$  on  $\nabla_x f(x)$  will be used to indicate with respect to which variables the derivative is being taken, so if  $y$  contains a subset of the variables,  $\nabla_y f(x)$  will denote the gradient with respect to these variables. If, in addition,  $f$  is twice-continuously differentiable ( $f \in C^2$ ), we define the *Hessian* matrix (or Hessian for short) of  $f$  to be the  $n$  by  $n$  matrix-valued function,  $\nabla_{xx} f(x)$ ,<sup>13</sup> whose  $(i,j)$ th entry is  $\partial^2 f(x)/\partial x_i \partial x_j$ . As the derivatives are continuous, the Hessian is necessarily symmetric. Once again, the suffix  $xx$  on  $\nabla_{xx} f(x)$  is used to indicate with respect to which variables the derivatives are being taken, and if  $y$  and  $z$  are subsets of the variables,  $\nabla_{yz} f(x)$  will denote the  $\dim y$  by  $\dim z$  submatrix of  $\nabla_{xx} f(x)$  corresponding to the variables  $y$  and  $z$ .

If  $c(x) \in C^1$  is a vector-valued function from  $\mathbb{R}^n$  to  $\mathbb{R}^m$ , we define its *Jacobian* matrix (or Jacobian for short) to be the  $m$  by  $n$  matrix-valued function  $\nabla_x c(x)$ , whose  $(i,j)$ th entry is  $\partial c_i(x)/\partial x_j$ , or, in matrix notation,

$$\nabla_x c(x) = \left( \begin{array}{c} \nabla_x c_1(x) \cdots \nabla_x c_m(x) \end{array} \right)^T.$$

Note that the Hessian of  $f(x)$  is the Jacobian of  $\nabla_x f(x)$ .

Taylor's theorem provides an important relationship between the values of a continuously differentiable function evaluated at two points. While this theorem may be stated in more generality, we shall find that the following variants suffice for our purposes.

**Theorem 3.1.1** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $f : \mathcal{S} \rightarrow \mathbb{R}$  is continuously differentiable throughout  $\mathcal{S}$ . Then, if the segment  $x + \theta s \in \mathcal{S}$  for all

<sup>11</sup>Having said this, we should note that relatively few researchers have found uses for third- and higher order derivatives, but this situation may change.

<sup>12</sup>Some authors call a function whose range is  $\mathbb{R}$  a functional, but we shall stick to the more general term throughout.

<sup>13</sup>Some authors prefer the notation  $\nabla^2 f(x)$ , but we prefer to avoid the confusion this brings with the Laplacian operator  $\nabla^2$ , which maps from  $\mathbb{R}$  to  $\mathbb{R}$ .

$\theta \in [0, 1]$ ,

$$f(x + s) = f(x) + \langle \nabla_x f(x + \alpha s), s \rangle$$

for some  $\alpha \in [0, 1]$ .

**Theorem 3.1.2** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $f : \mathcal{S} \rightarrow \mathbb{R}$  is twice-continuously differentiable throughout  $\mathcal{S}$ . Then, if the segment  $x + \theta s \in \mathcal{S}$  for all  $\theta \in [0, 1]$ ,

$$f(x + s) = f(x) + \langle \nabla_x f(x), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} f(x + \alpha s) s \rangle$$

for some  $\alpha \in [0, 1]$ .

There is also a version of Theorem 3.1.1 appropriate for vector-valued functions.

**Theorem 3.1.3: Integral mean value theorem** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $F : \mathcal{S} \rightarrow \mathbb{R}^m$  is continuously differentiable throughout  $\mathcal{S}$ . Then, if the segment  $x + \theta s \in \mathcal{S}$  for all  $\theta \in [0, 1]$ ,

$$F(x + s) = F(x) + \int_0^1 \nabla_x F(x + \alpha s) s d\alpha.$$

Notice that in this case, unlike in Theorem 3.1.1, we cannot conclude that there is an  $\alpha \in [0, 1]$  for which  $F(x + s)$  and  $F(x) + \nabla_x F(x + \alpha s) s$  coincide.

Having to rely on the values of derivatives at the unknown values  $x + \alpha s$  in Theorems 3.1.1 and 3.1.2, or on the unknown integral term in Theorem 3.1.3, can be a nuisance. We can avoid this if our functions satisfy stronger conditions.

Let  $\mathcal{X}$  and  $\mathcal{Y}$  be open sets, let  $F : \mathcal{X} \rightarrow \mathcal{Y}$ , and suppose that  $\|\cdot\|_{\mathcal{X}}$  and  $\|\cdot\|_{\mathcal{Y}}$  are norms on these sets. Then we say that such a function is *Lipschitz continuous* at  $x \in \mathcal{X}$  if there is a  $\gamma(x)$  such that

$$\|F(z) - F(x)\|_{\mathcal{Y}} \leq \gamma(x) \|z - x\|_{\mathcal{X}}$$

for all  $z \in \mathcal{X}$ . Notice that the Lipschitz constant  $\gamma(x)$  depends upon both  $x$  and the norms  $\|\cdot\|_{\mathcal{X}}$  and  $\|\cdot\|_{\mathcal{Y}}$ . The function is *locally Lipschitz continuous* near  $x$  if the set  $\mathcal{X}$  above is an open ball around  $x$ . It is Lipschitz continuous in  $\mathcal{X}$  if the constant is independent of  $x$ , that is, if there is a  $\gamma$  such that

$$\|F(z) - F(x)\|_{\mathcal{Y}} \leq \gamma \|z - x\|_{\mathcal{X}}$$

for all  $x, z \in \mathcal{X}$ ; it is locally Lipschitz in a set  $\mathcal{L}$  if it is locally Lipschitz near every  $x$  in  $\mathcal{L}$ . A locally Lipschitz function on an open set  $\mathcal{L}$  is differentiable almost every-

where<sup>14</sup> in  $\mathcal{L}$ —this is known as Rademacher’s theorem. The sum of a finite number of Lipschitz continuous functions in  $\mathcal{X}$  is Lipschitz continuous in  $\mathcal{X}$ . If the functions are also bounded in  $\mathcal{X}$ , the product of these functions is also Lipschitz continuous there. For functions with Lipschitz continuous derivatives we have the following variants of Theorems 3.1.1–3.1.3.

**Theorem 3.1.4** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $f : \mathcal{S} \rightarrow \mathbb{R}$  is continuously differentiable throughout  $\mathcal{S}$ . Suppose further that  $\nabla_x f(x)$  is Lipschitz continuous at  $x$ , with Lipschitz constant  $\gamma(x)$  in some appropriate vector norm. Then, if the segment  $x + \theta s \in \mathcal{S}$  for all  $\theta \in [0, 1]$ ,

$$|f(x + s) - m(x + s)| \leq \frac{1}{2}\gamma(x)\|s\|^2,$$

where

$$m(x + s) = f(x) + \langle \nabla_x f(x), s \rangle.$$

**Theorem 3.1.5** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $f : \mathcal{S} \rightarrow \mathbb{R}$  is twice-continuously differentiable throughout  $\mathcal{S}$ . Suppose further that  $\nabla_{xx} f(x)$  is Lipschitz continuous at  $x$ , with Lipschitz constant  $\gamma(x)$  in some appropriate vector norm and its induced matrix norm. Then, if the segment  $x + \theta s \in \mathcal{S}$  for all  $\theta \in [0, 1]$ ,

$$|f(x + s) - m(x + s)| \leq \frac{1}{6}\gamma(x)\|s\|^3,$$

where

$$m(x + s) = f(x) + \langle \nabla_x f(x), s \rangle + \frac{1}{2}\langle s, \nabla_{xx} f(x)s \rangle.$$

**Theorem 3.1.6** Let  $\mathcal{S}$  be an open subset of  $\mathbb{R}^n$ , and suppose  $F : \mathcal{S} \rightarrow \mathbb{R}^m$  is continuously differentiable throughout  $\mathcal{S}$ . Suppose further that  $\nabla_x F(x)$  is Lipschitz continuous at  $x$ , with Lipschitz constant  $\gamma(x)$  in some appropriate vector norm and its induced matrix norm. Then, if the segment  $x + \theta s \in \mathcal{S}$  for all  $\theta \in [0, 1]$ ,

$$\|F(x + s) - M(x + s)\| \leq \frac{1}{2}\gamma(x)\|s\|^2,$$

where

$$M(x + s) = F(x) + \nabla_x F(x)s.$$

These results indicate that the errors that result when approximating  $f(x + s)$  by its

---

<sup>14</sup>Except on a set of Lebesgue measure zero.

*first-order Taylor approximation*

$$f(x + s) \approx f(x) + \langle \nabla_x f(x), s \rangle$$

and its *second-order Taylor approximation*

$$f(x + s) \approx f(x) + \langle \nabla_x f(x), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} f(x)s \rangle$$

are of the order of  $\|s\|^2$  and  $\|s\|^3$ , respectively, whenever  $f$  has sufficiently Lipschitz continuous derivatives. Similarly, the error that occurs when a Lipschitz continuously differentiable function  $F(x + s)$  is replaced by its first-order Taylor approximation

$$F(x + s) \approx F(x) + \nabla_x F(x)s$$

is of the order of  $\|s\|^2$ .

### Notes and References for Subsection 3.1.2

See Gruver and Sachs (1980) and Dennis and Schnabel (1983) for useful summaries of Taylor's theorem and its consequences. Rademacher's theorem is given, for example, by Rockafellar (1983, Theorem 2.2).

### 3.1.3 Convexity

Convexity plays a central role in the mathematical theory of optimization, and many beautiful results arise when the problem obeys convexity assumptions. From a practical point of view, convexity is often hard to detect unless the problem is very special (for instance, a linear program<sup>15</sup>), and this is perhaps why many practitioners feel that the study of convexity is overrated from a global perspective. Nonetheless, globally nonconvex problems are sometimes convex in a neighbourhood of a solution; thus there are powerful reasons to study convexity, and we certainly make use of such results in this book.

A subset  $\mathcal{C}$  of  $\mathbb{R}^n$  is said to be *convex* if for any points  $x$  and  $y$  in  $\mathcal{C}$  and  $\theta \in [0, 1]$ , the point  $x + \theta(y - x)$  is also in  $\mathcal{C}$ ; the point  $x + \theta(y - x)$  is said to be a convex combination of  $x$  and  $y$ . Thus a convex set has no “holes”, nor can its boundary “bend back on itself”. By convention, the empty set is also convex. Examples of convex and nonconvex sets are given in Figure 3.1.1. The intersection of two (or more) convex sets is also convex. Perhaps the most useful convex set, from our perspective, is the region defined by a finite or infinite set of linear equations  $\langle a_i, x \rangle = b_i$  or inequalities  $\langle a_i, x \rangle \geq b_i$ . If  $\mathcal{S}$  is any subset of  $\mathbb{R}^n$ , another useful convex set is the *convex hull* of  $\mathcal{S}$ ,  $\text{co}\{\mathcal{S}\}$ , which is defined to be the intersection of all convex sets containing  $\mathcal{S}$ . The convex hull consists of all convex combinations of elements of  $\mathcal{S}$ .

For future reference, we formalize the basic assumption on the convexity of the set  $\mathcal{C}$  as follows.

---

<sup>15</sup>A linear program is the minimization of a linear function, subject to a set of linear constraints.

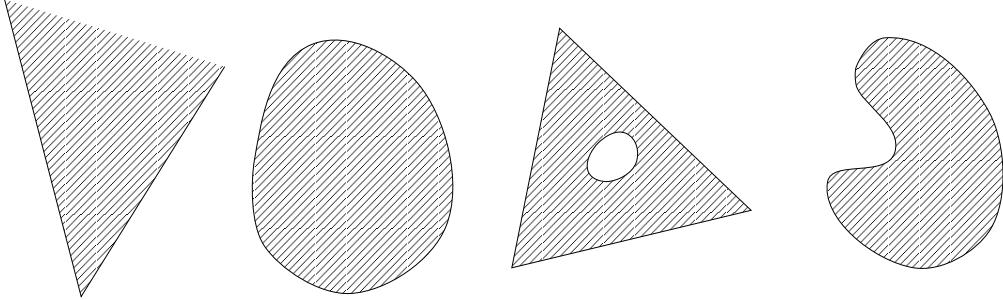


Figure 3.1.1: The two sets on the left are convex, while the two on the right are nonconvex.

**AC.2** The set  $\mathcal{C}$  is nonempty, closed, and convex.

There are two further properties of closed convex sets that are extremely important. The first is the *separating hyperplane theorem*. This says that if  $\mathcal{C}$  is a closed, convex set, and if  $y \notin \mathcal{C}$ , then there is a hyperplane  $\langle a, x \rangle = b$  such that  $\langle a, y \rangle > b$  and  $\langle a, x \rangle < b$  for all  $x \in \mathcal{C}$ . The second result relates to the minimum distance

$$\min_{x \in \mathcal{C}} \|x - y\| \quad (3.1.1)$$

from a closed convex set  $\mathcal{C}$  to a point  $y \notin \mathcal{C}$ . It turns out that the minimizer  $x_*$  of the least-distance problem (3.1.1) is unique and that  $x_*$  solves the problem (3.1.1) if and only if

$$\langle x - x_*, y - x_* \rangle \leq 0 \quad (3.1.2)$$

for all  $x \in \mathcal{C}$ .

Given a convex subset  $\mathcal{S}$  of  $\mathbb{R}^n$ , a function  $f : \mathcal{S} \rightarrow \mathbb{R}$  is said to be *convex* if

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

for all  $x, y \in \mathcal{S}$  and  $\theta \in [0, 1]$ . That is, the function always lies below its linear interpolant. We illustrate this in Figure 3.1.2. Such an  $f$  is *strictly convex* if

$$f(\theta x + (1 - \theta)y) < \theta f(x) + (1 - \theta)f(y)$$

for all  $x \neq y \in \mathcal{S}$  and  $\theta \in (0, 1)$ . We also say that  $f(x)$  is concave if  $-f(x)$  is convex. A positive linear combination of convex functions is convex. It is trivial to show that the subset of  $\mathcal{S}$  for which a convex  $c(x) \leq 0$  is a convex set. The minimization of a convex objective  $f(x)$  subject to a set of convex inequalities  $c_i(x) \leq 0$ , or more generally the minimization of such an  $f$  over a convex set (satisfying AC.2), is known as a *convex programming* problem.

Differentiable convex functions have a number of interesting features. If  $f(x) \in C^1$  is convex, it follows that

$$f(y) \geq f(x) + \langle y - x, \nabla_x f(x) \rangle.$$

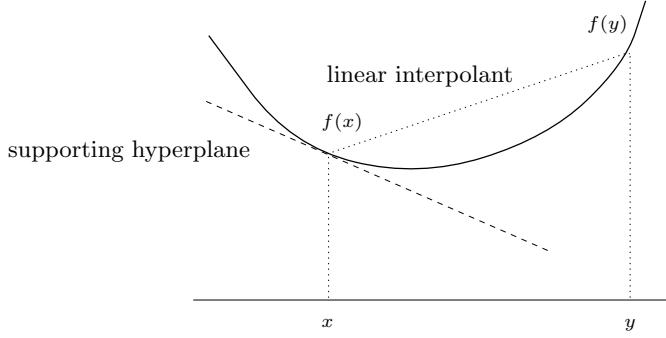


Figure 3.1.2: A convex function, its supporting hyperplane at  $x$ , and the linear interpolant passing through the points  $(x, f(x))$  and  $(y, f(y))$ .

Thus  $f(y)$  lies on or above its first-order Taylor approximation  $f(x) + \langle y - x, \nabla_x f(x) \rangle$ ; this linearization is known as the *supporting hyperplane* of  $f(x)$ . Once again, this is illustrated in Figure 3.1.2. It also follows that

$$\langle y - x, \nabla_x f(y) \rangle \geq \langle y - x, \nabla_x f(x) \rangle$$

and thus that the slope of  $f(x)$  does not decrease along any line; if  $f$  is strictly convex, the slope along any line must increase. When  $f(x) \in C^2$  is convex, its Hessian matrix is positive semidefinite for all  $x \in \mathcal{S}$ . If we define the *curvature* of  $f$  at  $x \in \mathcal{S}$  along the direction  $d$  by

$$\frac{\langle d, \nabla_{xx} f(x)d \rangle}{\|d\|^2},$$

we see that the curvature of a convex function is always nonnegative at any  $x$  and for all directions  $d$ . The converse is also true, namely, that if the Hessian matrix of  $f \in C^2$  is positive semidefinite throughout  $\mathcal{S}$ , or if the curvature of  $f$  at every  $x \in \mathcal{S}$  along any direction  $d$  is nonnegative, then  $f$  is convex there. If the Hessian is positive definite throughout  $\mathcal{S}$ , or the curvature positive in  $\mathcal{S}$  along any direction, then  $f$  is strictly convex there; however, the converse is not true.<sup>16</sup>

### Notes and References for Subsection 3.1.3

Once again, see Rockafellar (1970) or Luenberger (1969) for more details. Later in the book, we shall need the concepts of positive and negative curvature. If  $f \in C^2$ , the vector  $d$  is a *direction of positive curvature* at  $x$  if  $\langle d, \nabla_{xx} f(x)d \rangle > 0$ , and  $f$  is said to have *positive curvature* in this direction. Similarly,  $d$  is a *direction of negative curvature* at  $x$  if  $\langle d, \nabla_{xx} f(x)d \rangle < 0$ , and  $f$  is said to have *negative curvature* in this direction.

<sup>16</sup>Consider, for instance, the strictly convex function  $f(x) = x^4$  whose Hessian at the origin is only positive semidefinite.

### 3.1.4 Nonsmooth Functions

Of course, not all continuous functions have derivatives. Such functions are known as nonsmooth or nondifferentiable functions. Fortunately, in many cases it is still possible to define, and to use, suitable generalizations.

The one-sided *directional derivative*,  $f'_d(x)$ , of  $f$  at  $x$  along  $d$  is defined to be

$$f'_d(x) \stackrel{\text{def}}{=} \lim_{t \searrow 0} \frac{f(x + td) - f(x)}{t}.$$

If  $f$  is convex and  $f(x)$  is finite, this derivative exists and  $f'_{-d}(x) \leq f'_d(x)$ ; when  $f$  is continuously differentiable in some neighbourhood of  $x$ ,  $f'_d(x) = \langle \nabla_x f(x), d \rangle$ . The *subdifferential*,  $\partial f(x)$ , of  $f$  at  $x$  is the set

$$\partial f(x) = \{g \in \mathbb{R}^n \mid \langle g, d \rangle \leq f'_d(x) \text{ for all } d \in \mathbb{R}^n\},$$

while each member of the subdifferential is known as a *subgradient*. It then immediately follows that if  $f$  is continuously differentiable in some neighbourhood of  $x$ ,  $\partial f(x) = \{\nabla_x f(x)\}$ . Moreover, if  $f$  is convex and  $f(x)$  is finite, the subdifferential may more usefully be written as

$$\partial f(x) = \{g \in \mathbb{R}^n \mid f(x) + \langle g, d \rangle \leq f(x + d) \text{ for all } d \in \mathbb{R}^n\}. \quad (3.1.3)$$

Of course, not all functions are convex, and it is easy to construct nonconvex examples that do not have one-sided directional derivatives. To cope with this possibility, we define the *generalized directional derivative*,  $f_d^o(x)$ , of  $f$  at  $x$  along  $d$  as

$$f_d^o(x) \stackrel{\text{def}}{=} \lim_{t \searrow 0} \sup_{y \rightarrow x} \frac{f(y + td) - f(y)}{t}.$$

If  $f$  is locally Lipschitz continuous with constant  $\gamma$  near  $x$ , then its generalized directional derivative exists in every direction, and  $|f_d^o(x)| \leq \gamma \|d\|$ . So long as the one-sided directional derivative exists along  $d$ , it follows trivially that  $f'_d(x) \leq f_d^o(x)$ . As before, if  $f$  is continuously differentiable in some neighbourhood of  $x$ ,  $f_d^o(x) = \langle \nabla_x f(x), d \rangle$ . We may also define the *generalized gradient*,  $\partial f(x)$ , of  $f$  at  $x$  as the set

$$\partial f(x) = \{g \in \mathbb{R}^n \mid \langle g, d \rangle \leq f_d^o(x) \text{ for all } d \in \mathbb{R}^n\},$$

and the generalized gradient of a continuously differentiable function is simply  $\{\nabla_x f(x)\}$ . The generalized gradient of a function, which is locally Lipschitz continuous near  $x$ , is a compact, convex set, and the generalized directional derivative along  $d$  satisfies

$$f_d^o(x) = \max_{g \in \partial f(x)} \langle g, d \rangle. \quad (3.1.4)$$

This is important, as it is often possible to describe the convex set  $\partial f(x)$  in terms of a finite set of generators, and hence to evaluate  $f_d^o(x)$  by solving the optimization problem on the right-hand side of (3.1.4). An extremely useful result due to Clarke is that if  $f$  is locally Lipschitz near  $x$ , the generalized gradient may be generated as

$$\partial f(x) = \text{co} \left\{ \lim_{x_i \rightarrow x} \nabla_x f(x_i) \mid x_i \notin \Omega_f \right\}, \quad (3.1.5)$$

where  $\Omega_f$  is the set (with measure zero) of points at which Rademacher's theorem implies that  $f$  is not differentiable.

Finite-valued convex functions defined on open sets are Lipschitz continuous, and their subdifferential and generalized gradient coincide. In general, we say that a function is *regular* at  $x$  if  $f'_d(x)$  exists for all  $d$  and  $f'_d(x) = f_d^o(x)$ . For Lipschitz continuous functions, the requirement is simply that one-sided and generalized directional derivatives coincide. The function is regular on  $\mathcal{X}$  if it is regular for each  $x$  in  $\mathcal{X}$ .

A most important special case results when  $h$  is a regular, locally Lipschitz continuous function on a subset  $\mathcal{C}$  of  $\mathbb{R}^m$ , and  $f$  and  $c$  are continuously differentiable functions from  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathbb{R}$  and  $\mathcal{C}$ , respectively. For then the composite function  $f(x) + h(c(x))$  is locally Lipschitz and regular on  $\mathcal{X}$ . In this case we have the generalized chain rule

$$\begin{aligned}\partial[f(x) + h(c(x))] &= \nabla_x f(x) + (\nabla_x c(x))^T \partial h(c(x)) \\ &\equiv \{\nabla_x f(x) + (\nabla_x c(x))^T y \mid y \in \partial h(c(x))\}.\end{aligned}\quad (3.1.6)$$

Of particular importance is the case where  $h$  is a convex function on an open convex subset  $\mathcal{C}$  of  $\mathbb{R}^m$ , which falls into the above category.

### Notes and References for Subsection 3.1.4

See Rockafellar (1970, Part 5), Clarke (1983, Chapter 2), and Fletcher (1987a, Chapter 14) for further details. The result (3.1.5) is given, in a slightly more general form, by Clarke (1983, Theorem 2.5.1). Regular functions are sometimes also known as quasi-differentiable functions (see Womersley, 1982).

### 3.1.5 Geometry

While real sets of constraints are usually defined by a finite or infinite number of equations and inequalities, it is sometimes more useful to view the set of feasible points from a geometric, rather than algebraic, viewpoint. The advantage is that the feasible region is then independent of the coordinate system used, and this can sometimes lead to a deeper understanding of the underlying properties of optimization algorithms. The basic geometric building blocks we use are cones.

A subset  $\mathcal{C}$  of  $\mathbb{R}^n$  is a *cone* (with its vertex at the origin<sup>17</sup>) if it is closed under positive scalar multiplication, that is, if  $\theta x \in \mathcal{C}$  whenever  $x \in \mathcal{C}$  and  $\theta > 0$  (see Rockafellar, 1970, p. 13). See Figure 3.1.3. The most important cones for optimization applications are *convex cones*, which are simply sets that are both convex and cones, and are thus simply sets that are closed under addition and positive scalar multiplication. Trivial examples are subspaces of  $\mathbb{R}^n$ . The intersection of an arbitrary collection of convex cones is a convex cone. Notice, though, that in general, cones may be open or closed sets (or both).

---

<sup>17</sup>In general, a cone with vertex  $y$  is a translation  $y + \mathcal{C}$  of a cone with vertex at the origin. All cones are assumed to be with vertices at the origin unless otherwise stated.

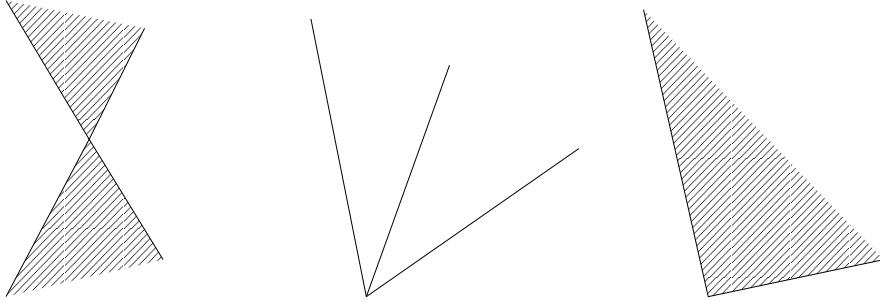


Figure 3.1.3: Three cones. The rightmost is handy because it is convex.

Given a cone  $\mathcal{K}$ , we define its *polar* as

$$\mathcal{K}^0 \stackrel{\text{def}}{=} \{y \in \mathbb{R}^n \mid \langle y, u \rangle \leq 0 \text{ for all } u \in \mathcal{K}\}.$$

It is easy to verify that  $\mathcal{K}^0$  is also a cone, and that  $(\mathcal{K}^0)^0 = \mathcal{K}$  whenever  $\mathcal{K}$  is a nonempty closed convex cone. We illustrate this in Figure 3.1.4.

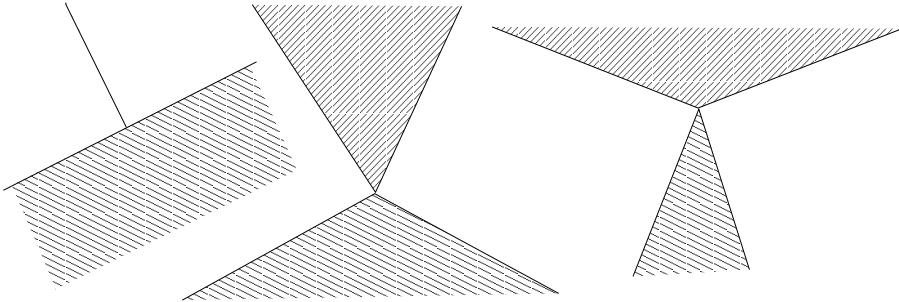


Figure 3.1.4: Three convex cones and their polars.

Let  $\mathcal{X}$  be a closed, convex set. Two cones play a special role in the theory of constrained optimization, the normal and tangent cones. The *normal* cone of  $\mathcal{X}$  at  $x \in \mathcal{X}$  is defined to be the set

$$\mathcal{N}(x) \stackrel{\text{def}}{=} \{y \in \mathbb{R}^n \mid \langle y, u - x \rangle \leq 0 \text{ for all } u \in \mathcal{X}\}.$$

The *tangent* cone of  $\mathcal{X}$  at  $x \in \mathcal{X}$  is the polar of the normal cone at the same point, that is,

$$\mathcal{T}(x) \stackrel{\text{def}}{=} \mathcal{N}(x)^0 = \text{cl}\{\theta(u - x) \mid \theta \geq 0 \text{ and } u \in \mathcal{X}\},$$

where  $\text{cl}\{\mathcal{S}\}$  denotes the closure of the set  $\mathcal{S}$  (see Section 3.1.1). These two cones are illustrated in Figure 3.1.5. Notice that if  $x$  lies in the interior of  $\mathcal{X}$ ,  $\mathcal{N}(x) = 0$  and  $\mathcal{T}(x) = \mathbb{R}^n$ .

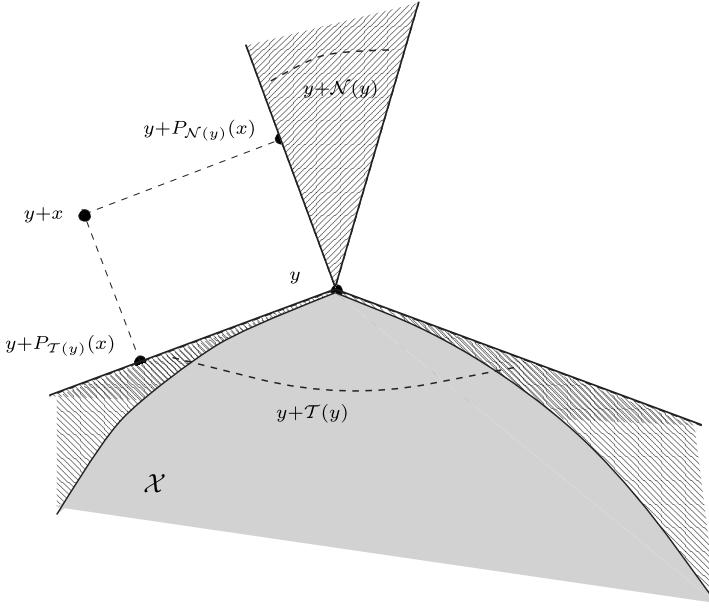


Figure 3.1.5: The normal and tangent cones of  $\mathcal{X}$  at  $y$ , and the Moreau decomposition of  $x$  relative to  $y$ .

A most important decomposition results from these two cones. We define the *projection* of the vector  $x \in \mathbb{R}^n$  onto  $\mathcal{X}$ ,  $P_{\mathcal{X}}(x)$ , as the unique minimizer of the problem

$$\min_{y \in \mathcal{X}} \|y - x\|_2.$$

The *Moreau decomposition* of  $x$  relative to  $y$  is given by the identity

$$x = P_{\mathcal{T}(y)}(x) + P_{\mathcal{N}(y)}(x), \quad (3.1.7)$$

which is valid for all  $x \in \mathbb{R}^n$  and all  $y \in \mathcal{X}$ . We illustrate this decomposition in Figure 3.1.5. We conclude this brief discussion of the properties of cones by noting that

$$P_{\mathcal{X}}(y + x) = P_{\mathcal{X}}(y) \text{ whenever } x \in \mathcal{N}(P_{\mathcal{X}}(y)), \quad (3.1.8)$$

a property which can also be verified by analysing Figure 3.1.5 and proved using the facts that, for all  $y$ , the vector  $y - P_{\mathcal{X}}(y)$  belongs to  $\mathcal{N}(P_{\mathcal{X}}(y))$  and that this latter cone is convex.

### Notes and References for Subsection 3.1.5

A full discussion of the geometry of cones is given by, for instance, Rockafellar (1970). The properties of projection operators are considered by, for example, Zarantonello (1971) and Luenberger (1969). Moreau (1962) was responsible for the decomposition that bears his name.

## 3.2 Optimality Conditions

In this book we shall be interested in finding local optima. The reason for this is simply that practical methods for finding global optima are, at present, too expensive in all but the most specialized of cases.<sup>18</sup> More significantly, unless very strong assumptions are made about the functions that define the optimization problem in question, characterizations of global minima are almost never possible,<sup>19</sup> so even if one is fortuitously found, we may never know. This is, of course, extremely disappointing, particularly as nonglobal local optima are of no interest for some problems.

We shall be concerned with the optimization problem of minimizing an *objective* function  $f(x)$  of  $n$  real variables  $x$ , where  $x$  is constrained to lie within a closed region  $\mathcal{C}$  of *feasible* points. This feasible region may be the whole of  $\mathbb{R}^n$ , in which case the problem is effectively *unconstrained*, or it may be a subset of  $\mathbb{R}^n$ , in which case the problem is *constrained*.

A feasible point  $x_*$  is a *local* minimizer of  $f(x)$  if there is an open neighbourhood  $\mathcal{N}$  of  $x_*$  such that  $f(x_*) \leq f(x)$  for all  $x \in \mathcal{C} \cap \mathcal{N}$ . The minimizer is strong or *strict* if there is an open neighbourhood  $\mathcal{N}$  of  $x_*$  such that  $f(x_*) < f(x)$  for all  $x \neq x_* \in \mathcal{C} \cap \mathcal{N}$ , it is *isolated* if there is no other local minimizers within some open neighbourhood of  $x_*$  and it is a *global* minimizer if  $f(x_*) \leq f(x)$  for all  $x \in \mathcal{C}$ . The value of  $f$  corresponding to a minimizer is a *minimum*.<sup>20</sup>

While these definitions are quite intuitive, they are of little practical use, as they do not tell us either how to find a minimizer or how to recognize one if we (accidentally or otherwise) stumble upon it. The rest of the book is devoted to the first of these issues, but in this section we consider the second. Our main tools will be, as usual, calculus and linear algebra, and our aim is to recast the abstract definitions of local optimality in terms of readily verifiable tests based upon derivatives or generalized derivatives. We shall not prove any of the results in this section, as they are established in most textbooks on optimization, and must refer the interested reader elsewhere for the details.

## Notes and References for Section 3.2

We recommend the discussions on optimality conditions for differentiable problems given in the books by Fiacco and McCormick (1968), Mangasarian (1979), Gill, Murray, and Wright (1981), and Fletcher (1981), and again recommend Fletcher (1981) as a good introduction to the more general case where derivatives may not exist.

---

<sup>18</sup>We do not mean here to decry the many attempts that have been made, particularly as the global minimizer is sometimes the only value of practical interest. We are merely reflecting on the state of implemented general-purpose algorithms for problems in more than, say, 10 variables.

<sup>19</sup>A notable exception is the  $\ell_2$ -norm trust-region subproblem considered in Section 7.2.

<sup>20</sup>Note that a strict minimizer can only fail to be isolated if there is an infinite number of minimizers in a neighbourhood.

### 3.2.1 Differentiable Unconstrained Problems

We first consider the unconstrained minimization of a differentiable function  $f(x)$ . We have the following optimality criterion.

**Theorem 3.2.1** Suppose that  $f \in C^1$ , and that  $x_*$  is a local minimizer of  $f(x)$ . Then

$$\nabla_x f(x_*) = 0. \quad (3.2.1)$$

The requirement (3.2.1) is known as the first-order (necessary) optimality condition for an unconstrained minimizer of  $f(x)$ . Any point  $x_*$  that satisfies (3.2.1) is said to be a *first-order critical* or *first-order stationary* point of  $f$ . We now turn to higher order derivatives. We shall use the following blanket assumption throughout the book.

**AF.1** The objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is twice-continuously differentiable on  $\mathbb{R}^n$ .

**Theorem 3.2.2** Suppose that AF.1 holds and that  $x_*$  is a local minimizer of  $f(x)$ . Then (3.2.1) holds and the objective function's Hessian at  $x_*$  is positive semidefinite; that is,

$$\langle s, \nabla_{xx} f(x_*) s \rangle \geq 0 \text{ for all } s \in \mathbb{R}^n. \quad (3.2.2)$$

The requirements (3.2.1) and (3.2.2) are known as the second-order (necessary) optimality conditions for an unconstrained minimizer of  $f(x)$ , and any point  $x_*$  that satisfies both conditions is said to be a *second-order critical point* of  $f$ . Of course, (3.2.2) merely says that the Hessian matrix of the objective function is necessarily positive semidefinite at a minimizer.

Finally, we consider under what conditions a suspected minimizer turns out to be strict.

**Theorem 3.2.3** Suppose that AF.1 holds, that  $x_*$  satisfies the condition (3.2.1), and that additionally the objective function's Hessian at  $x_*$  is positive definite; that is,

$$\langle s, \nabla_{xx} f(x_*) s \rangle > 0 \text{ for all } s \neq 0 \in \mathbb{R}^n. \quad (3.2.3)$$

Then  $x_*$  is a strict, isolated local minimizer of  $f(x)$ .

The conditions (3.2.1) and (3.2.3) are known as the second-order sufficiency conditions

for  $x_*$  to be an unconstrained minimizer of  $f(x)$ . The condition (3.2.3) requires that the Hessian matrix of the objective function be positive definite at  $x_*$ .

Notice that it follows immediately that necessary conditions for  $x_*$  to be a *maximizer* of  $f \in C^2$  is that it be a first-order critical point and that  $\nabla_{xx}f(x_*)$  be *negative* definite, that is, that  $-\nabla_{xx}f(x_*)$  be positive definite. When  $x_*$  is first-order critical, but  $\nabla_{xx}f(x_*)$  is indefinite,  $x_*$  is known as a *saddle* point.

### 3.2.2 Differentiable Constrained Problems

We now turn to the problem where the variables are subjected to constraints. To be specific, we shall consider the general constrained optimization problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \text{ for } i \in \mathcal{E} \\ & && \text{and} \quad c_i(x) \geq 0 \text{ for } i \in \mathcal{I}, \end{aligned} \tag{3.2.4}$$

where  $\mathcal{E}$  and  $\mathcal{I}$  are disjoint sets of the indices of *equality* and *inequality* constraints, and where  $f$  and the  $c_i$  map  $\mathbb{R}^n$  into  $\mathbb{R}$ . Although simplifications result for specific classes of constraints, we prefer to identify these when and where necessary. We let  $c$  be the vector whose components are the  $c_i$ , and write  $c_{\mathcal{E}}$  and  $c_{\mathcal{I}}$  to be the vectors whose components  $c_i$  occur in  $\mathcal{E}$  and  $\mathcal{I}$ , respectively; the ordering within these vectors is not important.

At any feasible point  $x$ , we define the *active set*

$$\mathcal{A}(x) = \{i \in \mathcal{E} \cup \mathcal{I} \mid c_i(x) = 0\}.$$

Note that  $\mathcal{A}(x) = \emptyset$  when  $x$  lies in the relative interior of the feasible set, if it exists. The set of constraints whose indices lie in the active set are said to be *active*, or *binding* while the remainder are *inactive*. The feasible set can then be partitioned into *faces*,  $\mathcal{F}_{\mathcal{S}}$ , which are the sets of feasible points having the same active set  $\mathcal{S}$ . These concepts are illustrated in Figure 3.2.1.

The aim of any algorithm for constrained optimization must surely be ultimately to discover which constraints are active at the sought minimizer. For then the inactive constraints may be discarded, and attention fixed on the equality and active ones. As we shall see later in the book, different algorithms do this in different ways, some by brute force and some by stealth.

In order to make any useful statements about constrained minima, we need to exclude pathological geometric cases. We do this by assuming that so-called *constraint qualifications* hold at a suspected minimum  $x_*$ . As the exact form of these constraint qualifications will not be used in this book, we need not be specific about them. However, we simply mention that close to a suspected minimum, we would expect any nonlinear constraint to be reasonably approximated by its first-order Taylor approximation. The first-order constraint qualification merely requires that these linear approximations characterize all feasible perturbations about  $x_*$ , while the second-order

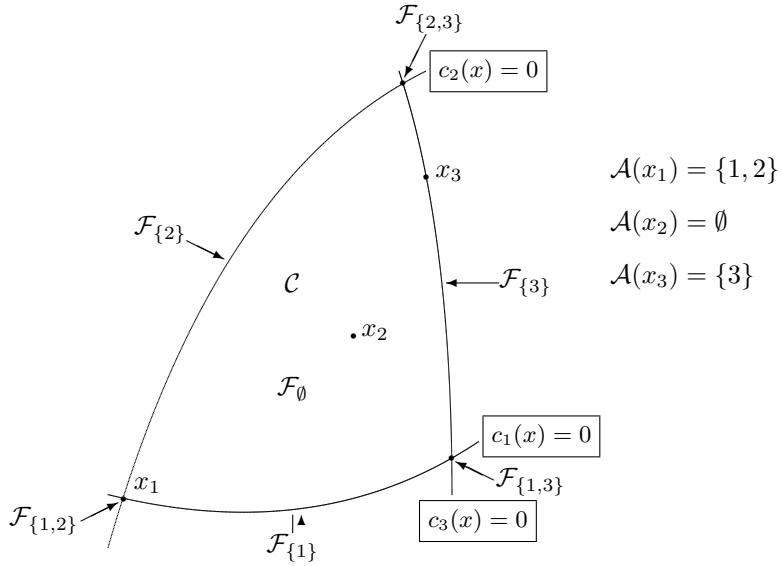


Figure 3.2.1: Faces and active sets.

one insists that perturbations that keep strongly<sup>21</sup> active constraints strongly active be completely characterized by their corresponding linearizations being forced to be active. Two cases when these constraint qualifications automatically hold are when all constraints are linear and when all the gradients of the active constraints are linearly independent. The usual example in which the constraint qualification fails to hold is when the solution is on the boundary at a cusp point. In any case, we shall use the following assumptions.

**AO.1** A first-order constraint qualification holds at  $x_*$ .

**AO.2** A second-order constraint qualification holds at  $x_*$ .

We may now state the required classification results. We start by considering the analog of Theorem 3.2.1 for the constrained case.

**Theorem 3.2.4** Suppose that  $f, c \in C^1$ , and that  $x_*$  is a local solution of (3.2.4). Then, provided that AO.1 holds, there exists a vector of *Lagrange multipliers*  $y_*$  such that

$$\nabla_x f(x_*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_*]_i \nabla_x c_i(x_*), \quad (3.2.5)$$

<sup>21</sup>Roughly, a constraint is strongly active if no small perturbations to the problem make it inactive as the solution shifts. It corresponds to a constraint with strictly positive Lagrange multipliers; see Theorem 3.2.4.

$$c_i(x_*) = 0 \text{ for all } i \in \mathcal{E}, \quad (3.2.6)$$

$$c_i(x_*) \geq 0 \text{ and } [y_*]_i \geq 0 \text{ for all } i \in \mathcal{I}, \quad (3.2.7)$$

$$\text{and } c_i(x_*)[y_*]_i = 0 \text{ for all } i \in \mathcal{I}. \quad (3.2.8)$$

The requirements stated in this theorem are known as the first-order (necessary) optimality, or commonly the Karush–Kuhn–Tucker (KKT), conditions for a solution of (3.2.4). Any point  $x_*$  that satisfies (3.2.5)–(3.2.8) is said to be a *first-order critical*, or KKT, point for the problem (3.2.4). The last of these conditions, (3.2.8), is known as the *complementary slackness* condition, while the first, (3.2.5), requires that the gradient of the Lagrange function or *Lagrangian*

$$\ell(x, y) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} y_i c_i(x),$$

taken with respect to the variables  $x$ , vanish at a KKT point. If we define  $N$  as a matrix whose columns form a basis for the null-space

$$\mathcal{N} = \{s \in \mathbb{R}^n \mid \langle s, \nabla_x c_i(x_*) \rangle = 0 \text{ for all } i \in \mathcal{A}(x_*)\} \quad (3.2.9)$$

of the space of gradients of the constraints active at  $x_*$ , then (3.2.5) is equivalent to requiring that the *reduced gradient*

$$N^T \nabla_x f(x_*) = 0. \quad (3.2.10)$$

This identity is somewhat more convenient as it is independent of the actual values of the Lagrange multipliers. We next turn to conditions involving second derivatives. The following blanket assumption will be used throughout.

**AC.1** The constraint functions  $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$  are twice-continuously differentiable on  $\mathbb{R}^n$ .

**Theorem 3.2.5** Suppose that AF.1 and AC.1 hold, and that  $x_*$  is a local solution of (3.2.4). Then, provided that AO.1 and AO.2 hold, there exists a vector of Lagrange multipliers  $y_*$  such that (3.2.5)–(3.2.8) hold and

$$\langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle \geq 0 \text{ for all } s \in \mathcal{N}_+, \quad (3.2.11)$$

where

$$\mathcal{N}_+ = \left\{ s \in \mathbb{R}^n \mid \begin{array}{l} \langle s, \nabla_x c_i(x_*) \rangle = 0 \text{ for all } i \in \mathcal{E} \cup \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j > 0\} \\ \text{and } \langle s, \nabla_x c_i(x_*) \rangle \geq 0 \text{ for all } i \in \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j = 0\} \end{array} \right\}. \quad (3.2.12)$$

The requirements (3.2.5)–(3.2.8) and (3.2.11) are known as the second-order (necessary) optimality conditions for a solution of (3.2.4), and any point  $x_*$  that satisfies these conditions is said to be a *strong second-order critical point* for the problem (3.2.4). Conversely, suppose we strengthen these conditions very slightly, and assume that

**AO.3** the conditions (3.2.5)–(3.2.8) and

$$\langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle > 0 \text{ for all } s \in \mathcal{N}_+ \quad (s \neq 0) \quad (3.2.13)$$

hold at  $(x_*, y_*)$ , where  $\mathcal{N}_+$  is defined by (3.2.12).

We then have the following converse to Theorem 3.2.5.

**Theorem 3.2.6** Suppose that AF.1 and AC.1 hold, and that AO.3 holds at  $(x_*, y_*)$ . Then  $x_*$  is a strict local solution of (3.2.4).

The conditions (3.2.5)–(3.2.8) and (3.2.13) are known as the second-order sufficiency conditions for  $x_*$  to be a local solution of (3.2.4).

One further very useful result is that, under certain circumstances, slightly perturbing the data for a problem with a strict local minimizer results in a nearby strict local minimizer for the perturbed problem. To state our result, we need to make the following further assumptions.

**AO.1b** The Jacobian of active constraint gradients,  $\nabla_x c_{\mathcal{A}(x_*)}(x_*)$ , is of full rank.

**AO.4** The set

$$\left\{ i \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_i = 0 \right\} = \emptyset. \quad (3.2.14)$$

The condition (3.2.14) is known as *strict* complementary slackness and implies that  $\mathcal{N} = \mathcal{N}_+$ . The assumptions AO.1b, AO.3, and AO.4 together imply that  $y_*$  is unique.

**Theorem 3.2.7** Consider the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x, p) \quad \text{subject to} \quad c_{\mathcal{E}}(x, p) = 0 \quad \text{and} \quad c_{\mathcal{I}}(x) \geq 0, \quad (3.2.15)$$

where  $p$  is a set of parameters, and (3.2.15) is (3.2.4) when  $p = 0$ . Suppose that the second derivatives of  $f$  and  $c$  are jointly continuous as functions of  $x$  and  $p$  and that assumptions AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$ . Then there is some open set  $\mathcal{P}$  containing the origin and some open neighbourhood  $\mathcal{X} \times \mathcal{Y}$  of  $(x_*, y_*)$  for which the continuous function  $(x(p)_*, y(p)_*)$ , with  $(x(0)_*, y(0)_*) = (x_*, y_*)$ , also satisfies AO.1b, AO.3, and AO.4 for all  $p \in \mathcal{P}$ . Furthermore  $x(p)_*$  is the only strict local minimizer of (3.2.15) in  $\mathcal{X}$ ,  $y(p)_*$  is the unique vector of Lagrange multipliers at this point, and  $\mathcal{A}(x(p)_*) = \mathcal{A}(x_*)$ .

### Notes and References for Subsection 3.2.2

The KKT conditions were for a long time attributed to Kuhn and Tucker (1951) but have more recently been found to have been stated in the earlier master's thesis of Karush (1939). A large number of alternative constraint qualifications have been proposed, some weaker than others. See Gould and Tolle (1972), Mangasarian (1979), and the papers quoted therein. While such assumptions are necessary, it is fortunate that practical examples where they are violated are rare or specific to special problem classes, and to all intents and purposes they may be (and normally are) ignored in the design of general-purpose algorithms.

Lagrange multipliers are often known as *dual variables*, since for convex problems they may be interpreted as variables for an appropriate "dual" problem (see, for instance, Mangasarian, 1979). Here, we shall choose to use the terms Lagrange multiplier and dual variable interchangeably, without any implication that the underlying problem is actually convex.

The second-order necessary conditions given here are those given by Fletcher (1981, Section 9.3). Significantly weaker (and hence less satisfactory) conditions are given by, for instance, Fiacco and McCormick (1968, Section 2.2) and Gill, Murray, and Wright (1981, Section 3.4), which are equivalent to requiring that the critical point  $(x_*, y_*)$  be strictly complementary, that is, that AO.4 holds, and thus that  $\mathcal{N}_+ = \mathcal{N}$ . While such an assumption is realistic for linear programming—all linear programs have such solutions (see Wright, 1997, p. 28) and many interior-point methods find one—it frequently does not hold for nonlinear programs. The advantage of assuming (3.2.14) is that the second-order optimality conditions reduce to checking that the Hessian of the Lagrangian is positive (semi)definite on the manifold (3.2.9) rather than in the cone (3.2.12). The former is relatively straightforward to check by examining the definiteness of the *reduced Hessian*

$$N^T \nabla_{xx} \ell(x_*, y_*) N,$$

while the latter is a much more difficult combinatorial problem (see Murty and Kabadi, 1987, for details). We shall call the requirement that

$$\langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle \geq 0 \text{ for all } s \in \mathcal{N} \quad (3.2.16)$$

a *weak* second-order necessary condition, while the assumption that

$$\langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle > 0 \text{ for all } s \in \mathcal{N} \ (s \neq 0)$$

is a *strong* second-order sufficient condition; a point that satisfies first-order and weak second-order necessary conditions is known as a *weak second-order critical point*. That (3.2.16) is weaker than (3.2.11) is clear once one realizes that the weak condition is satisfied by the *maximizer* of the quadratic programming problem

$$\min_{\substack{x \in \mathbb{R}^n \\ x \geq 0}} -\|x\|_2^2,$$

while (3.2.5)–(3.2.8) and the strong condition are, together, both necessary and sufficient for local optimality of general quadratic programs (see Contesse, 1980; Mangasarian, 1980; and Borwein, 1982).

As we have said, checking weak second-order necessary and strong second-order sufficiency conditions is straightforward so long as the null-space matrix  $N$  is available and an inertia-revealing factorization of the reduced Hessian is feasible. If this is not the case, it is still

possible to examine whether these conditions hold by examining the inertia of the matrix

$$K(x_*, y_*) = \begin{pmatrix} \nabla_{xx}\ell(x_*, y_*) & A^T(x_*) \\ A(x_*) & 0 \end{pmatrix},$$

where  $A^T(x)$  is the matrix whose columns are the active constraint gradients  $\nabla_x c_i(x)$ ,  $i \in \mathcal{A}(x)$ . The strong second-order sufficiency condition holds if and only if  $K(x_*, y_*)$  has precisely  $n$  positive eigenvalues, while the weak second-order necessary condition holds if and only if  $K(x_*, y_*)$  has precisely  $\text{rank}(A(x_*))$  negative eigenvalues. These and other related conditions are due to Chabriac and Crouzeix (1984) and Gould (1985), and may be checked by forming an appropriate symmetric indefinite factorization of  $K$ .

Theorem 3.2.7 is due to Robinson (1974). This result has been strengthen by Fiacco (1976) to show that  $(x(p)_*, y(p)_*)$  are differentiable so long as  $f, c \in C^2$  as functions of  $(x, p)$ . See Fiacco (1983) for further generalizations.

### 3.2.3 Convex Programming

We have already defined the minimization of a convex objective  $f(x)$  over a convex set  $\mathcal{C}$  satisfying AC.2 as a convex programming problem. In this context, convexity has one extremely useful attribute and one important consequence: *every local minimizer of a convex program is a global minimizer*, and, if the problem (3.2.4) is convex (i.e.,  $c_i(x)$ ,  $i \in \mathcal{E}$  are linear and  $f(x)$  and  $-c_i(x)$ ,  $i \in \mathcal{I}$  are convex) and differentiable (i.e.,  $f, c \in C^1$ ), the conditions (3.2.5)–(3.2.8) are sufficient for global optimality. If, in addition,  $f$  is strictly convex, the global minimizer is unique. Thus, for certain readily identifiable convex programs (such as linear and convex quadratic programming<sup>22</sup>), the search for a global solution reduces to the hunt for a local one. Notice that convexity does not ensure that there is a (finite) global minimum, merely that there is no more than one local minimum; to ensure more than this, we would require additional assumptions such as that  $\mathcal{C}$  be nonempty and compact.

Another useful property of convex programming problems is that they behave well when the problem is perturbed. We express this property by considering a family of convex problems whose objective function and feasible set are continuously parametrized by a vector  $y$  in some domain  $\mathcal{D}$  of  $\mathbb{R}^\ell$ .

**Theorem 3.2.8** Assume that  $\mathcal{C}$  is a continuous point-to-set mapping from  $\mathcal{D} \subseteq \mathbb{R}^\ell$  into  $\mathbb{R}^n$  such that the set  $\mathcal{C}(y)$  is convex for each  $y \in \mathcal{D}$ . Assume also that one is given a real-valued function  $f(x, y)$ , which is defined and continuous on  $\mathbb{R}^n \times \mathcal{D}$  and convex for each fixed  $y$ . Then, the real-valued function  $f_*$  defined on  $\mathcal{D}$  by

---

<sup>22</sup>The standard definition of a convex quadratic program is the minimization of a quadratic function whose Hessian is positive semidefinite subject to a set of linear constraints. However, this is really unnecessarily restrictive. A more reasonable definition would be that the quadratic function is convex over the feasible set. So, for example, the problem “minimize  $\frac{1}{2}x^T Hx + c^T x$  subject to  $Ax = b$ ” would be considered convex if  $N^T HN$  is positive semidefinite where  $AN = 0$ .

$$f_*(y) = \inf_{x \in \mathcal{C}(y)} f(x, y)$$

and the solution set mapping  $x_*$  defined on  $\mathcal{D}$  by

$$x_*(y) = \{x \in \mathcal{C}(y) \mid f(x, y) = f_*(y)\}$$

are both continuous on  $\mathcal{D}$ .

We illustrate this result in Figure 3.2.2, where the solutions of two neighbouring convex programming problems are shown. (The two problems, depending on the parameters  $y_1$  and  $y_2$ , are drawn in thick and thin lines, respectively.)

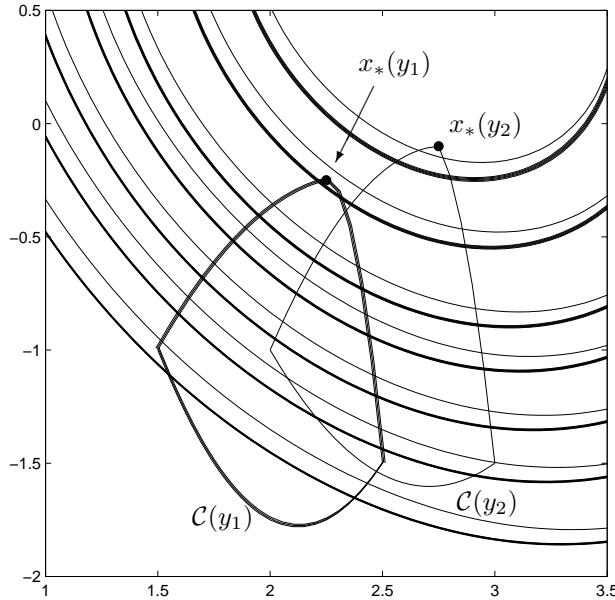


Figure 3.2.2: The continuity of the solutions of convex programs.

If the constraint set  $\mathcal{C}$  is convex in the sense of AC.2, but  $f(x)$  is any (not necessarily) convex function, the first-order optimality conditions may be expressed in an alternative, geometric form.

**Theorem 3.2.9** Suppose that  $\mathcal{C}$  satisfies AC.2, that  $f$  is continuously differentiable in  $\mathcal{C}$ , and that  $x_*$  is a first-order critical point for the minimization of  $f$  over  $\mathcal{C}$ . Then, provided that AO.1 holds,

$$-\nabla_x f(x_*) \in \mathcal{N}(x_*).$$

The advantage of this characterization over the KKT conditions is simply that the normal cone,  $\mathcal{N}(x_*)$ , is independent of the coordinate system used, and thus avoids the use of basis-dependent Lagrange multipliers.

We now consider the case in which the feasible set  $\mathcal{C}$  is described by a finite number of smooth constraints, which we formally state for future reference as follows.

**AC.7** The feasible set  $\mathcal{C}$  is the intersection of a finite number of sets of the form

$$\mathcal{C} = \bigcap_{i=1}^m \mathcal{C}_i, \quad \text{where } \mathcal{C}_i = \{x \in \mathbb{R}^n \mid c_i(x) \geq 0\},$$

where each  $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ .

The feasible set of Figure 3.2.1 is an example of a feasible set satisfying AC.2, AC.1, and AC.7.

Note that  $\mathcal{C}$  may be convex in the sense of AC.2, while some of the sets  $\mathcal{C}_i$  may not be. The interest in situations in which AC.1 and AC.7 hold, together with a first-order constraint qualification, is that the normal cone of  $\mathcal{C}$  at  $x$  may then be characterized in terms of the outward normals  $-\nabla_x c_i(x)$  of the active constraints at  $x$ .

**Theorem 3.2.10** Assume that AO.1, AC.2, AC.1, and AC.7 hold. Then

$$\mathcal{N}(x) = \left\{ x \in \mathbb{R}^n \mid x = - \sum_{i \in \mathcal{A}(x)} \alpha_i \nabla_x c_i(x) \text{ with } \alpha_i \geq 0 \text{ for } i = 1, \dots, m \right\}.$$

Furthermore,  $\mathcal{N}(x)$  and  $\mathcal{T}(x)$  are continuous within each face of  $\mathcal{C}$ .

We conclude this overview of convex programming problems by restating once more the first-order optimality conditions, this time for problems in which the feasible set is defined as the intersection of some convex set  $\mathcal{C}_0$  and the set of all vectors satisfying explicit differentiable constraints.

**Theorem 3.2.11** Suppose that  $f, c \in C^1$  and that  $x_*$  is a local solution of the problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \text{ for } i \in \mathcal{E}, \\ & && c_i(x) \geq 0 \text{ for } i \in \mathcal{I}, \\ & \text{and} && x \in \mathcal{C}_0, \end{aligned}$$

where  $\mathcal{C}_0$  is a nonempty, closed, and convex subset of  $\mathbb{R}^n$ . Then, provided that AO.1 holds, there exists a vector of Lagrange multipliers  $y_*$  such that

$$-\nabla_x f(x_*) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_*]_i \nabla_x c_i(x_*) \in \mathcal{N}(x_*), \quad (3.2.17)$$

$$c_i(x_*) = 0 \text{ for all } i \in \mathcal{E}, \quad (3.2.18)$$

$$c_i(x_*) \geq 0 \text{ and } [y_*]_i \geq 0 \text{ for all } i \in \mathcal{I}, \quad (3.2.19)$$

$$\text{and } c_i(x_*)[y_*]_i = 0 \text{ for all } i \in \mathcal{I}, \quad (3.2.20)$$

where  $\mathcal{N}(x)$  is the normal cone to  $\mathcal{C}_0$  at  $x$ .

This expression of the first-order optimality conditions is illustrated in Figure 3.2.3 for a simple problem with only one equality and no inequality constraints. In this situation, the negative gradient of the objective is “deflated” by its component along the gradient of the constraint (see the dotted lines), and the resulting vector with the multiplier  $y_*$  approximately equal to 2 lies in the normal cone to the feasible set at the critical point (the dashed line), as prescribed by (3.2.17).

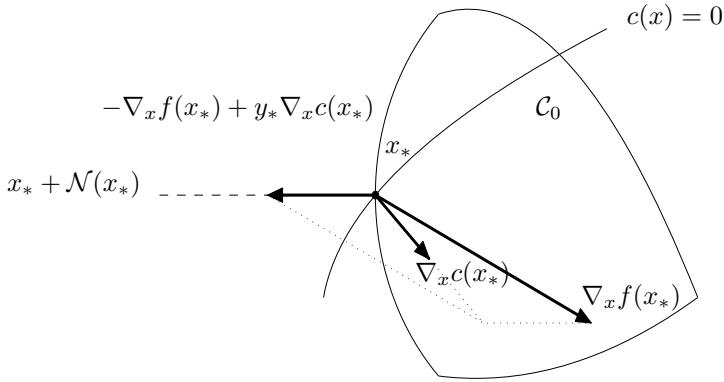


Figure 3.2.3: The situation of Theorem 3.2.11.

### Notes and References for Subsection 3.2.3

See Luenberger (1969) and Rockafellar (1970) for a detailed discussion of the material in this section. Theorem 3.2.8 is a well-known result of perturbation theory for mathematical programs and can be found, for instance, in Fiacco (1983, pp. 14–17).

### 3.2.4 Nonsmooth Problems

We finally consider the case of minimizing the continuous but nonsmooth function  $f(x)$ . In this case, we have the following optimality criterion.

**Theorem 3.2.12** Suppose that  $f$  is locally Lipschitz continuous near a local minimizer  $x_*$ . Then

$$0 \in \partial f(x_*), \quad (3.2.21)$$

or equivalently,

$$f_d^o(x_*) \equiv \max_{g \in \partial f(x_*)} \langle g, d \rangle \geq 0 \text{ for all } d \in \mathbb{R}^n. \quad (3.2.22)$$

Any point  $x_*$  that satisfies (3.2.21) is said to be a *first-order critical* point of  $f$ . When the function is regular, (3.2.22) implies that  $f'_d(x_*) \geq 0$ , while for irregular functions it is possible that there are directions for which the one-sided directional derivative is negative even at a local minimizer.

For composite functions  $f(x) + h(c(x))$  with locally Lipschitz  $h$  and differentiable  $f$  and  $c$ , we have the following corollary.

**Corollary 3.2.13** Suppose that  $h$  is a locally Lipschitz continuous function on a subset  $\mathcal{C}$  of  $\mathbb{R}^m$ , and  $f$  and  $c$  are continuously differentiable functions from  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathbb{R}$  and  $\mathcal{C}$ , respectively. Then if  $x_* \in \mathcal{X}$  is a local minimizer of the composite function  $f(x) + h(c(x))$ ,

$$0 \in \nabla_x f(x_*) + (\nabla_x c(x_*))^T \partial h(c(x_*)),$$

or equivalently,

$$f_d^o(x_*) \equiv \max_{y \in \partial h(c(x_*))} \langle d, \nabla_x f(x_*) + (\nabla_x c(x_*))^T y \rangle \geq 0 \text{ for all } d \in \mathbb{R}^n.$$

One can also obtain very general second-order necessary and sufficient conditions for local optimality that resemble the analogous results for smooth optimization, but where curvature conditions are replaced by appropriate generalizations of directional curvature. Since we shall not use these results in this book, we will not give further details. However, there is one special case we must cover, namely, that for the composite function  $f(x) + h(c(x))$  with convex  $h$  and differentiable  $f$  and  $c$ . In this case, suitable second-order sufficiency conditions are that

**AO.3n** the relationships

$$\nabla_x f(x_*) + (\nabla_x c(x_*))^T y_* = 0$$

and

$$\left\langle d, \left( \nabla_{xx} f(x_*) + \sum_{i=1}^n [y_*]_i \nabla_{xx} c_i(x_*) \right) d \right\rangle > 0 \quad (3.2.23)$$

hold for all  $d$  in

$$\mathcal{D} = \left\{ d \mid \max_{y \in \partial h(c(x_*))} \langle d, \nabla_x f(x_*) + (\nabla_x c(x_*))^T y \rangle = 0 \text{ and } \|d\| = 1 \right\} \quad (3.2.24)$$

at  $x_* \in \mathcal{X}$  and  $y_* \in \partial h(c(x_*)$ .

In this case, we have the following result.

**Theorem 3.2.14** Suppose that  $h$  is a convex function on a subset  $\mathcal{C}$  of  $\mathbb{R}^m$ , and  $f$  and  $c$  are twice-continuously differentiable functions from  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathbb{R}$  and  $\mathcal{C}$ , respectively. If there are  $x_* \in \mathcal{X}$  and  $y_* \in \partial h(c(x_*))$  such that AO.3n holds, then  $x_*$  is a strict, isolated local minimizer of  $f(x) + h(c(x))$ .

A necessary version of this theorem is also possible in which a local minimizer implies that, with simple inequality replacing the strict inequality, (3.2.23) holds under suitable regularity assumptions.

Further generalizations are possible. For instance, one might wish to minimize a nonsmooth function subject to a set of smooth or even nonsmooth constraints. Although we shall not consider such a general problem in this book, there is one special case that will occur, namely, that of minimizing a composite nonsmooth function subject to a single nonsmooth constraint of the form  $t(r(x)) \leq 0$ , where  $t$  is convex and  $r$  is smooth.

**Theorem 3.2.15** Suppose that  $h$  and  $t$  are convex functions on subsets  $\mathcal{C}$  of  $\mathbb{R}^m$  and  $\mathcal{T}$  of  $\mathbb{R}^p$ , respectively, and that  $f$ ,  $c$ , and  $t$  are continuously differentiable functions from  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathbb{R}$ ,  $\mathcal{C}$ , and  $\mathcal{T}$ , respectively. Then, if  $x_* \in \mathcal{X}$  is a local minimizer of the composite function  $f(x) + h(c(x))$  subject to the constraint  $t(r(x)) \leq 0$ , and if AO.1 holds, there are  $y_* \in \partial h(c(x_*))$ ,  $u_* \in \partial t(r(x_*))$ , and  $z_* \geq 0$  such that

$$\begin{aligned} \nabla_x f(x_*) + (\nabla_x c(x_*))^T y_* + z_* (\nabla_x r(x_*))^T u_* &= 0, \\ t(r(x_*)) &\leq 0 \text{ and } t(r(x_*))z_* = 0. \end{aligned} \quad (3.2.25)$$

Any point  $x_*$  that satisfies (3.2.25) is a first-order critical point. Suitable second-order optimality conditions are easily derived (the reader can probably guess their form) under further constraint qualifications, but will not be used. We note that the constraint qualification AO.1 is automatically satisfied in many important cases, such as when  $t(r(x)) = \|x\|$  for any given norm, or if  $t(r(x)) = \|r(x)\|_1$ .

### Notes and References for Subsection 3.2.4

See Clarke (1983, Section 2.3) for further details. The second-order conditions stated here are due to Fletcher (1987a, Section 14.2). More general conditions are given, for example, by Huang and Ng (1994). The case where constraints are involved is covered by Fletcher and Watson (1980) and Fletcher (1987a, Section 14.6).

## 3.3 Sequences and Basic Iterations

At the heart of almost all the methods we shall consider in this book lies an iteration. Normally, the problem we wish to solve is sufficiently complicated that a closed-form solution is unavailable, and we must, instead, resort to iterative procedures to estimate a solution. An iteration is a process by which an estimate,  $x_k$ , of the solution to the problem is replaced by (what we hope is) a better estimate,  $x_{k+1}$ . In this section, we consider means of assessing the convergence of a sequence of iterates  $\{x_k\}$ , as well as describing the king of all iterations, Newton's method.

### 3.3.1 Sequences

Throughout the book, we shall encounter sequences of numbers and their limits. Suppose that  $\{y_k\}$  and  $\{z_k\}$ ,  $k = 0, 1, \dots$ , are two sequences of positive real numbers. Then we say that  $y_k$  is asymptotically bounded by  $z_k$ , and write  $y_k = O(z_k)$ , if

$$y_k \leq \kappa_1 z_k$$

for some constant  $\kappa_1$  and all  $k \geq 0$  sufficiently large. If  $y_k = O(z_k)$  and  $z_k = O(y_k)$ , then the two sequences are asymptotically similar, and we write  $y_k = \Theta(z_k)$ . The sequence  $\{y_k\}$  is asymptotically smaller than  $\{z_k\}$  if

$$\lim_{k \rightarrow \infty} \frac{y_k}{z_k} = 0,$$

and we then write  $y_k = o(z_k)$ . When the sequences are vectors or matrices, it suffices to compare their norms.

The rate at which a sequence approaches a limit is also of interest. Suppose that the vector sequence  $\{x_k\}$  converges to  $x_*$ , that is, that

$$\lim_{k \rightarrow \infty} \|x_k - x_*\| = 0.$$

Then there are a number of different ways of measuring the speed or *rate* of convergence. The simplest is the Q-rate.<sup>23</sup> The sequence is said to converge *Q-linearly* if there is a constant  $\kappa_2 \in [0, 1)$  such that

$$\|x_{k+1} - x_k\| \leq \kappa_2 \|x_k - x_*\|$$

---

<sup>23</sup>Q stands for quotient.

for all  $k$  larger than some  $k_0 \geq 0$ . Note that this statement depends on the actual choice of the norm (because  $\kappa_2$  must be in  $[0, 1)$ ). The rate is *Q-superlinear* if

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

while it is *Q-quadratic* if

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

as  $k$  tends to infinity. These two latter properties are norm independent. Another measure of the speed of convergence is the R-rate.<sup>24</sup> The sequence  $\{x_k\}$  is said to converge with an R-rate of  $p$  if  $\|x_k - x_*\|$  is bounded by another sequence which converges to zero at a Q-rate of  $p$ . For instance,  $\{x_k\}$  is R-linearly convergent if

$$\|x_k - x_*\| \leq \omega_k,$$

where  $\omega_k$  converges Q-linearly to zero. Notice that one strong advantage of a sequence that is Q- rather than R-linearly convergent is that the norm of the error  $\|x_k - x_*\|$  for the former must eventually decrease monotonically, while that for the latter may increase from time to time so long as the trend is downward.

### Notes and References for Subsection 3.3.1

An interesting history of the  $O$ ,  $o$ , and  $\Theta$  notation is given by Knuth (1976). Our definition of the R-order of convergence is from Dennis and Schnabel (1983). A detailed description of the relationships between Q- and R-orders of convergence is provided by Ortega and Rheinboldt (1970).

### 3.3.2 Newton's Method

Probably the best-known iterative method of all is Newton's method. This is an iterative procedure for finding a root of the differentiable nonlinear system of equations

$$F(x) = 0,$$

where  $F : \mathcal{X} \rightarrow \mathcal{Y}$ , and  $\mathcal{X}$  and  $\mathcal{Y}$  are open subsets of  $\mathbb{R}^n$ . The method is summarized as Algorithm 3.3.1.

**Algorithm 3.3.1: Newton's method**

**Step 0: Initialization.** An initial point  $x_0$  is given. Set  $k = 0$ .

---

<sup>24</sup>R stands for root.

**Step 1: Compute the Newton step.** Find the step  $s_k$  as a solution of the system of linear equations

$$\nabla_x F(x_k) s_k = -F(x_k). \quad (3.3.1)$$

**Step 2: Update the estimate of the solution.** Set

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

increment  $k$  by 1, and return to Step 1.

The linear equations (3.3.1) are known as the *Newton equations* or *system*, while the solution  $s_k$  is the *Newton step* or *correction*.

The method is particularly important because most other methods aspire to it. It is simple. More importantly, when it works, it is (asymptotically) very fast. Of the many convergence results that have been proved, the following is typical.

**Theorem 3.3.1** Suppose that  $\nabla_x F$  is Lipschitz continuous throughout the open, convex set  $\mathcal{X}$  and that the iterates generated by Algorithm 3.3.1 have a limit point  $x_*$  for which  $F(x_*) = 0$  and for which  $\nabla_x F(x_*)$  is nonsingular. Then  $\{x_k\}$  and  $\{F(x_k)\}$  converge at a Q-quadratic rate.

Notice what the result says and does not say. When convergence to a root occurs, the convergence will normally be (asymptotically) fast. But there is no implication that convergence will occur, and in practice this is a fundamental difficulty. Indeed, without such a drawback, it is unlikely that this book would be necessary. Nevertheless, under stronger conditions it is possible to infer that the iterates converge to a root.

**Theorem 3.3.2** Suppose that  $\nabla_x F$  is Lipschitz continuous throughout the open, convex set  $\mathcal{X}$ . Then so long as  $\nabla_x F(x_0)$  is nonsingular and  $\nabla_x F(x_0)^{-1} F(x_0)$  is sufficiently small, Algorithm 3.3.1 is well defined and converges to a point  $x_*$  for which  $F(x_*) = 0$  at an R-quadratic rate.

It is perhaps best to consider Newton's method as an idealized algorithm, in the sense that Algorithm 3.3.1 may not be convergent from arbitrary points. The hope is always, however, that once the iterates of a more sophisticated algorithm enter the region of convergence of Newton's method, the algorithm will revert to the Newton iteration. Other drawbacks of the method are that the Jacobian of  $F$  is required, that a linear system has to be solved at each iteration, and that this system may be ill-

conditioned (see Section 4.3.1) or even singular. We shall address all of these issues as the book progresses.

### Notes and References for Subsection 3.3.2

The reader should refer to Ortega and Rheinboldt (1970) for a thorough description of the local convergence properties of Newton's method and its relatives. A useful summary is provided by Dennis and Schnabel (1983). Theorem 3.3.2 is due to Kantorovich (1948).

### 3.3.3 Forcing Functions

In our later developments, we will often need the concept of a *forcing function*. We will say that  $\phi(\cdot)$  is a forcing function if it is a continuous function from the set of nonnegative reals  $\mathbb{R}_+$  into itself with the property that

$$\phi(x) = 0 \text{ if and only if } x = 0.$$

Examples of forcing functions of  $x$  are  $x$  itself and  $x^2$ . The products or the sums of sets of forcing functions are forcing functions, as are their minimum or maximum.

### Notes and References for Subsection 3.3.3

The terminology “forcing functions” appears in Ortega and Rheinboldt (1970).

# Chapter 4

---

---

## Basic Linear Algebra

---

---

We now turn to the other cornerstone of optimization: linear algebra and the numerical solution of linear systems.

### 4.1 Linear Equations in Optimization

We are fortunate that the linear systems of equations that underpin most optimization algorithms almost always involve matrices that are real and symmetric.<sup>25</sup> As we have already seen in Section 2.2, this implies that we need only be concerned with matrices with real coefficients and with real eigenvalues. It is fair to say that the generic linear system in optimization involves matrices of the form

$$\begin{pmatrix} H & A^T \\ A & -D \end{pmatrix}, \quad (4.1.1)$$

where  $H$  and  $D$  are symmetric,  $A$  is rectangular (or occasionally square) with fewer rows than columns, and typically  $D$  is diagonal and positive definite and has small coefficients. This implies that the generic optimization matrix (4.1.1) is indefinite. More fundamentally, optimization matrices of the form (4.1.1) frequently have the important property that their inertia is  $(n, m, 0)$ , where  $n$  and  $m$  are the dimensions of  $H$  and  $D$ , respectively; this property is related to the second-order optimality conditions we considered in Section 3.2.2.

There are two particularly important special cases. Firstly, when the system arises from unconstrained optimization,  $m = 0$  and the inertial condition is simply that  $H$ , the Hessian of the objective function, be positive definite. Secondly, when  $m$  constraints are present, it frequently happens that  $D = 0$  and  $H$  is the Hessian of the Lagrangian function, while  $A$  is the constraint Jacobian. The case in which  $D$  is nonzero normally corresponds to a perturbation of the optimality conditions, such as may occur in penalty

<sup>25</sup>The simplex method for linear programming is the most notable exception, but even here the equations may be interpreted as simplifications of symmetric systems.

and barrier methods (see Chapters 13 and 14), or to a separation of terms with different structures, as may happen for nonlinear least squares (see Section 4.1.2).

Thus we shall concentrate on linear systems involving real, symmetric matrices. This is not to say that unsymmetric matrices do not have roles to play—we shall see the importance of triangular and orthonormal matrices as tools for factorizing matrices—merely that the systems themselves are defined in terms of symmetric matrices.

### 4.1.1 Structure and Sparsity

Before we start discussing specific cases, there is one further crucial issue we must consider, namely, the structure of the matrix involved. For simplicity, we may categorize a matrix as dense, structured, or sparse. But, unlike other attributes we have considered, such as symmetry or type of coefficient, these classifications are entirely subjective, depending crucially on both how the matrices are to be used and the environment in which they are being used; as we shall see, a specific matrix may be sparse in some circumstances and dense in others.

A *dense* matrix is simply one in which little effort is made to exploit any structure it may have. Thus a diagonal matrix may be considered to be dense if the off-diagonal zeros are stored and manipulated. A *structured* matrix, on the other hand, is one in which much effort is made to exploit seen and unseen structure. An important class of structured matrices are *sparse*; that is, they have a (significant) number of zero coefficients. For instance, a diagonal matrix is structured—and sparse—if its off-diagonal zeros are ignored. But structured matrices need not be sparse. For example, if  $a$  is a vector of  $n$  (nonzero) values, the matrix  $I + aa^T$  has no nonzeros, but can nevertheless be inverted in roughly  $2n$  floating-point operations if considered as a structured matrix, as opposed to roughly  $\frac{1}{6}n^3$  if it is treated as dense. Moreover, just because a matrix has a large number of zero coefficients does not necessarily imply that it is sparse. In particular, every attempt to exploit these zeros may be thwarted, or the effort involved in trying to do so may outweigh the benefit—manipulating sparse matrices is typically more troublesome than treating matrices as dense. Many modern computer architectures are comfortable with dense matrices, because of the predictability of the data storage, whereas they may be handicapped by the irregularity of data for sparse matrices. Thus it is now common for matrices of modest order, say, in the hundreds, with a sizeable number of zero elements to be treated as dense, simply because it is more efficient to do so. But this also depends upon the (computing) environment: on machines with modest storage and memory it may be impossible to store or manipulate zero values, and thus only structured/sparse matrix processing is possible when the order is large.

We should also stress the difference here between direct (factorization) and iterative methods. The latter are normally always able to exploit the structure/sparsity of a matrix because the only operations involving the matrix are products with a succession of given vectors. Direct methods, as we shall see, are less dependent on the structure of the matrix than on the structure of its factors. This may be a handicap, as this

latter structure may not be known in advance.

### 4.1.2 Structured Systems

Symmetric structured matrices typically occur in the form

$$B = H + A^T D A,$$

where  $A$  may be rectangular. There are two main ways of exploiting this structure: either by seeing that this implies structure in the inverse of  $H$  or through augmentation.

Suppose that  $H$  and  $D$  are (easily) invertible. Then the Sherman–Morrison–Woodbury formula for the inverse of  $B$  is that

$$B^{-1} = H^{-1} - H^{-1} A^T (D^{-1} + AH^{-1}A^T)^{-1} AH^{-1},$$

so that the solution to  $Bx = b$  is

$$x = H^{-1}b - H^{-1}A^T(D^{-1} + AH^{-1}A^T)^{-1}AH^{-1}b. \quad (4.1.2)$$

This formula is then the basis of a method for finding  $x$  so long as we can factorize  $H$ ,  $D$ , and  $D^{-1} + AH^{-1}A^T$ , as all other terms in the above expression involve matrix–vector products with  $A$  and  $A^T$ . This method is usually used when  $H$  is sparse and  $A$  has only a few rows.

An alternative when  $H$  is not (easily) invertible is to observe that

$$b = Bx = Hx + A^T D Ax = Hx + A^T y,$$

where  $y = DAx$ . Thus  $x$  and the auxiliary vector  $y$  also satisfy the *augmented* system

$$\begin{pmatrix} H & A^T \\ A & -D^{-1} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}. \quad (4.1.3)$$

The coefficient matrix of this system is symmetric, indefinite, and larger than that of the original system, but may be sparse. Notice that we may recover (4.1.2) by eliminating the variables  $y$  in (4.1.3).

## 4.2 Special Matrices

The solution of nonsingular linear systems of equations  $Ax = b$  is probably the most fundamental problem in computational science, and one that is central to this book. The first of two main approaches to this problem is to decompose or *factorize*  $A$  into the product of two or more special matrices that are easily invertible. For, suppose we can factorize  $A$  as  $A = BC$ ; then we may solve  $Ax = b$  by first solving the auxiliary problem  $By = b$  to find  $y$ . We may subsequently recover  $x$  by solving  $Cx = y$ ; if  $A$  is nonsingular, then  $B$  and  $C$  will be also, so long as they are square. The essential idea then is to determine for which matrices we may easily extract solutions to associated linear equations, and to find factorizations that involve these key matrices. In this section, we consider the first of these issues.

### 4.2.1 Diagonal Matrices

A square matrix  $D$  is *diagonal* if  $d_{i,j} = 0$  for all  $i \neq j$ . Diagonal matrices hold a most elevated position in the matrix hierarchy. Not only are their eigenvalues clearly on display, but the solution of a nonsingular diagonal system of equations,  $Dx = b$ , is trivially available as  $x_i = b_i/d_i$ .

### 4.2.2 Triangular Matrices

A square matrix  $L$  is *lower triangular* if  $l_{i,j} = 0$  for all  $i < j$ , while  $U$  is *upper triangular* if  $u_{i,j} = 0$  for all  $i > j$ . A triangular matrix is *unit triangular* if each of its diagonal entries is 1. A triangular matrix is nonsingular if and only if all its diagonal entries are nonzero; the inverse of a nonsingular (unit) triangular matrix is (unit) triangular.

The solution of nonsingular triangular systems of equations is straightforward. Suppose  $L$  is nonsingular and lower triangular, and that we wish to find  $x$  so that  $Lx = b$ . The first equation involves only  $x_1$ , the next involves only  $x_2$  and the just-determined  $x_1$ , the next involves only  $x_3$  and the previously determined  $x_1$  and  $x_2$ , and so on. Thus the  $j$ th equation only involves the unknown  $x_j$  and the previously determined  $x_1, \dots, x_{j-1}$ , and we simply determine each component  $x_i$  one at a time starting with  $x_1$  and finishing with  $x_n$ . This process is known as *forward substitution* or *forward solution*. The solution of nonsingular upper triangular systems is just as easy, but now the solution is obtained in the opposite order, by simply determining  $x_n$  from the last equation, then  $x_{n-1}$  from  $x_n$  in the next-but-last, and finally  $x_1$  from the previously determined  $x_2, \dots, x_n$  in the first equation. For upper triangular matrices, the solution process is known as *back substitution* or *back solution*.

### 4.2.3 Orthonormal Matrices

A square matrix  $Q$  is said to be *orthonormal* if

$$Q^T Q = I. \quad (4.2.1)$$

This immediately reveals that

$$QQ^T = I \text{ and } Q^T = Q^{-1}.$$

It also follows trivially that the product of orthonormal matrices is orthonormal. Orthonormal matrices are particularly useful because the multiplication of any vector by an orthonormal matrix preserves its length (in the  $\ell_2$  norm). For, if  $y = Qx$ ,

$$\|y\|_2^2 = \langle Qx, Qx \rangle = \langle x, Q^T Qx \rangle = \langle x, x \rangle = \|x\|_2^2,$$

and thus  $\|y\|_2 = \|x\|_2$ . The principal use of orthonormal matrices is to transform a poor coordinate system into a useful one.

The simplest orthonormal matrices are *permutation* matrices, which are made up of permutations of the columns of the identity matrix. The simplest of these is  $I$

itself. If  $A$  is any square matrix and  $P$  is a permutation matrix, premultiplication by  $P$  permutes the rows of  $A$ , while postmultiplication permutes its columns.

Another useful class of orthonormal matrices is the *plane-rotation* or *Givens* matrices. A plane-rotation matrix  $P_{i,j,\theta}$  is defined by two indices  $i$  and  $j$  and an angle  $\theta$ , and is simply the matrix formed by replacing the  $i$ , $i$ th,  $i$ , $j$ th,  $j$ , $i$ th, and  $j$ , $j$ th entries of  $I$  by the quantities  $\cos \theta$ ,  $\sin \theta$ ,  $\sin \theta$ , and  $-\cos \theta$ , respectively.<sup>26</sup> Plane rotations are normally used to reduce an element of a given vector to zero. For instance, if we wish to reduce the  $i$ th entry of  $x$  to zero, premultiplication by  $P_{i,j,\theta}$  will achieve this aim if  $\theta = -\arctan x_i/x_j$ . Notice that the plane rotation affects the entry in position  $j$  as well as  $i$ , but leaves all other entries unchanged.

The final important class of orthonormal matrices is that of *plane-reflector* or *Householder* matrices. While plane rotations may be used to reduce a single entry of a vector to zero, plane reflectors are used to do the same to more than one entry simultaneously. A plane-reflector matrix is an orthonormal matrix of the form

$$P_w = I - \frac{2}{\langle w, w \rangle} w w^T$$

for some appropriate vector  $w$ .<sup>27</sup> In order to transform a vector  $x$  into a vector  $y$  with the same norm, we merely pick  $w = x - y$ . The same effect may be achieved, at increased cost, using a sequence of plane rotations.

The solution of orthonormal systems of equations is very straightforward. For if  $Q$  is orthonormal and we wish to find  $x$  so that  $Qx = b$ , it follows immediately from (4.2.1) that  $x = Q^T b$ .

## Notes and References for Subsection 4.2.3

Orthonormal matrices play a central role in numerical linear algebra because of their excellent numerical properties. Matrix problems are frequently simplified by applying sequences of simple orthonormal matrices such as plane rotations or reflections. Particular important applications are to the  $LQ$ ,  $QR$ , and complete orthonormal factorizations and to the solution of linear least-squares problems. See Section 4.4.1, Golub and Van Loan (1989, Chapter 6), Higham (1996, Chapters 18 and 19), Lawson and Hanson (1974), Björck (1996), or any other numerical linear algebra textbook for details.

### 4.2.4 Tridiagonal and Band Matrices

A matrix  $B$  is *tridiagonal* if  $b_{i,j} = 0$  for all  $|i - j| > 1$ . More generally, it is *banded* if  $|i - j| > k$  for some  $k < n$ ; the value  $k$  is the semibandwidth. Unlike triangular or orthonormal matrices, solving systems with band matrices usually requires an additional factorization (the inverse of a band matrix is generally full), but the factorization is often able to take account of the band structure of  $B$ . For instance, if  $B$  is banded,

---

<sup>26</sup>Strictly, this is a symmetric plane rotation. If, instead, the last two entries are  $-\sin \theta$  and  $\cos \theta$ , the plane rotation is unsymmetric.

<sup>27</sup>By convention, we define  $P_0 = I$ .

symmetric, and positive definite, then a factorization of the form  $B = T_1 T_2$  is possible in which  $T_1$  and  $T_2$  are triangular matrices whose bandwidth is no larger than that of  $B$ . The subsequent forward and back substitutions may clearly take this structure into account. Such a decomposition is a special case of the one we shall consider in Section 4.3.3.

## 4.3 Direct Methods for Solving Linear Systems

Having considered suitable building blocks for matrix factorization, we now turn to the factorization of  $A$  itself. Before considering such methods in detail, we must first mention two important considerations.

### 4.3.1 Stability in the Face of Rounding Errors

As the methods we consider are most likely to be implemented on computers that use finite-precision arithmetic, we need to be careful that our methods are as immune to the effects of computer rounding as possible, or at least to fully understand the limitations of the methods. The most popular technique for analysing methods for solving linear systems proceeds in two stages, the first based on fundamental properties of the system being solved, and the second on the properties of the solution method used.

The first stage is to consider what effects a perturbation of the data  $(A, b)$  has on the solution  $x$ . With this in mind, we suppose that

$$(A + \Delta A)(x + \Delta x) = b + \Delta b, \quad \text{where } Ax = b.$$

This immediately reveals that

$$\Delta x = (I + A^{-1} \Delta A)^{-1} A^{-1} (\Delta b - \Delta A x). \quad (4.3.1)$$

If we now suppose that

$$\|A^{-1} \Delta A\| < 1, \quad (4.3.2)$$

it can be shown that

$$\|(I + A^{-1} \Delta A)^{-1}\| \leq \frac{1}{1 - \|A^{-1} \Delta A\|}. \quad (4.3.3)$$

Taking norms of both sides of (4.3.1) and using the triangle inequality, (4.3.3), and the inequality

$$\frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|},$$

we conclude that

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \|A^{-1} \Delta A\|} \left( \frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right), \quad (4.3.4)$$

where

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

is known as the *condition number* for the solution of linear systems. Notice that while most of the terms on the right-hand side of (4.3.4) depend on the data perturbations, the condition number depends on the data itself, and thus is a characteristic of the problem. We say that the problem is ill-conditioned if  $\kappa(A)$  is large, and well-conditioned otherwise. Notice also that the condition number depends on the norm used (for the  $\ell_2$  norm,  $\kappa_2(A) = \sigma_{\max}[A]/\sigma_{\min}[A]$ ) and is always at least 1. Although the perturbation bound (4.3.4) may be extremely pessimistic for a given  $b$ , there is always at least one right-hand side for which the relative perturbation in the solution is close to the bound given on the right-hand sides of (4.3.4). It is worth reminding the reader that the bound (4.3.4) depends on the assumption (4.3.2). It is trivial to show that (4.3.2) holds so long as the relative perturbation in  $A$  is smaller than the reciprocal of the condition number.

The second stage is to use the perturbation bound (4.3.4) to show that a particular method gives a satisfactory approximation to the solution to  $Ax = b$ . To do this, it then suffices to show that the solution computed by a given method is the exact solution of a slightly perturbed system. This is known as a backward error analysis, and a method that passes this test is judged to be numerically stable. The exact details of such analyses are normally extremely intricate and method specific and focus on the rounding errors made when forming the factorization and when recovering the solution from  $b$  and the computed factors of  $A$ . Fortunately, the most popular methods, including those mentioned in this section, have all been subjected to rigorous error analyses and are (to varying degrees, but for all practical purposes) stable.

### Notes and References for Subsection 4.3.1

The development of backward error analysis and its application to many well-known matrix factorizations is due to Wilkinson (1963, 1965). Such analyses have been performed on more recent methods in the intervening years by a variety of authors. More subtle, componentwise error analyses often provide tighter bounds. See Golub and Van Loan (1989) and Higham (1996) for details.

### 4.3.2 Sparse Systems

When a system is sparse, the aim is clearly to make as much use as possible of the fact that certain coefficients are zero. As the ordering of the equations and the labelling of the unknowns are arbitrary, the main tool of sparse matrix methods is static or dynamic equation/variable reordering, with the aim of keeping the required work close to a minimum. Since our matrices are symmetric, we shall only be concerned with symmetric reordering.

As a simple example, consider the two symmetric positive definite matrices of order 4,

$$A_1 = \begin{pmatrix} x & . & . & . \\ x & x & & \\ x & & x & \\ x & & & x \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} x & & & . \\ & x & & . \\ & & x & . \\ x & x & x & x \end{pmatrix},$$

where the  $x$ 's indicate *entries* (values that are stored, regardless of their numerical value) and spaces indicate zeros which are not stored; the strict upper triangles would not be stored since the matrices are symmetric, but a *dot* indicates nonzero, unstored entries. Suppose we aim to decompose each matrix into the product of a lower triangular matrix and its transpose—this is the Cholesky factorization which we shall very shortly encounter. Then it can be seen that the resulting lower triangular matrices for our examples have the form

$$L_1 = \begin{pmatrix} x & & & \\ x & x & & \\ x & y & x & \\ x & y & y & x \end{pmatrix} \quad \text{and} \quad L_2 = \begin{pmatrix} x & & & \\ & x & & \\ & & x & \\ x & x & x & x \end{pmatrix},$$

where the  $y$ 's indicate *fill-ins*, that is, entries in positions that were created during the decomposition. It should then be clear that if  $A_2$  were simply a permuted version of  $A_1$ , we would prefer to factorize  $A_2$ , as its factors are cheaper to obtain and require less storage. Thus, in this (rather extreme) example, a sparse matrix method presented with  $A_1$  would first reorder its rows and columns so that it had the form  $A_2$  and then apply the decomposition—the reordering might only be performed implicitly to avoid data movement. Although general matrices are likely to fill in during factorization whatever permutation is applied, the goal of sparse matrix factorization is to limit the fill-in as much as possible.

The above example points to a general strategy: Firstly, permute the rows and columns of  $A$  with the aim of reducing the fill-in during the factorization, and then perform the factorization itself. The first phase is often called the *symbolic analysis*, while the second is referred to as the *numerical factorization*. The symbolic analysis is applied on the basis of the sparsity structure of  $A$  without regard to the numerical values of its entries. Although finding the ordering that leads to the least possible fill-in is a hard combinatorial problem,<sup>28</sup> there are good heuristics that aim to approach the least fill-in at an acceptable cost. For some classes of matrices, as we shall see, it is necessary to incorporate further permutations in the factorization phase to ensure a stable factorization. Unfortunately, in these cases, some of the good work achieved in the analysis phase may be undone by the subsequent extra stabilizing permutations.

The entire solution process may be summarized as finding a permutation matrix  $P$  so that  $A^P = PAP^T$  has low fill-in but may be stably decomposed, and then solving  $Ax = b$  by solving

$$A^P x^P = b^P, \quad \text{where} \quad b^P = Pb,$$

using the factors of  $A^P$  and subsequently recovering  $x = P^T x^P$ .

---

<sup>28</sup>It lies in the class of NP hard problems, for which it is thought that there is no polynomial algorithm.

### Notes and References for Subsection 4.3.2

A full description of sparse-matrix techniques is contained in the books of Duff, Erisman, and Reid (1986) and George and Liu (1981).

#### 4.3.3 Symmetric Positive Definite Matrices

Suppose that  $A$  is symmetric and positive definite. Then we may decompose  $A$  as

$$A = LL^T, \quad (4.3.5)$$

where  $L$  is a square lower triangular matrix. This is known as the *Cholesky* factorization of  $A$ . The factors are unique up to the signs of the columns of  $L$ . As the computation of the diagonal entries of  $L$  requires taking square roots, the alternative  $LDL^T$  factorization

$$A = LDL^T,$$

where  $L$  is now unit lower triangular and  $D$  is positive definite and diagonal, is sometimes preferred. If  $L^C$  is the lower triangular matrix from the Cholesky factorization of  $H$  and  $L^L$  is that from its  $LDL^T$  factorization, we may write  $L^C = D^{\frac{1}{2}}L^L$ .

The Cholesky factor  $L$  may be computed in a number of equivalent ways. One way is to compute it column by column. Suppose that

$$A = \begin{pmatrix} a_{11} & a_1^T \\ a_1 & A_{22} \end{pmatrix} \quad \text{and} \quad L = \begin{pmatrix} l_{11} & 0 \\ l_1 & L_{22} \end{pmatrix}. \quad (4.3.6)$$

Combining (4.3.5) and (4.3.6) and equating terms immediately reveals that

$$l_{11} = \sqrt{a_{11}}, \quad l_1 = a_1/l_{11}, \quad \text{and} \quad L_{22}L_{22}^T = A_{22} - l_1 l_1^T. \quad (4.3.7)$$

This then gives us the first column of  $L$ , and the remaining factor  $L_{22}$  is simply the Cholesky factor of the so-called Schur complement<sup>29</sup>  $A_{22} - l_1 l_1^T$ . Thus the factor may be found one column at a time. A second way is to compute it row by row. Suppose that we have the factorization  $A_{kk} = L_{kk}L_{kk}^T$  of the first  $k$  by  $k$  submatrix  $A_{kk}$  of  $A$ , and that we wish to compute its next row. Then on writing

$$A_{k+1k+1} = \begin{pmatrix} A_{kk} & a_{k+1} \\ a_{k+1}^T & a_{k+1k+1} \end{pmatrix} \quad \text{and} \quad L_{k+1k+1} = \begin{pmatrix} L_{kk} & 0 \\ l_{k+1}^T & l_{k+1k+1} \end{pmatrix},$$

using the first  $k + 1$  by  $k + 1$  submatrices of (4.3.5) and equating terms, we have that

$$L_{kk}l_{k+1} = a_{k+1} \quad \text{and} \quad l_{k+1k+1} = \sqrt{a_{k+1k+1} - l_{k+1}^T l_{k+1}}. \quad (4.3.8)$$

---

<sup>29</sup>If  $A$  is invertible and

$$K = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

the matrix  $D - CA^{-1}B$  is known as the *Schur complement* of  $A$  in  $K$ .

Thus the off-diagonal part of the  $(k + 1)$ st row,  $l_{k+1}$ , of the required factor may be found by solving a lower triangular system of equations (see Section 4.2.2), and  $l_{k+1k+1}$  is then recovered by taking the square root of the pivot<sup>30</sup> entry  $a_{k+1k+1} - l_{k+1}^T l_{k+1}$ .

Both approaches indicate the limitations of the Cholesky factorization. In particular, the method will fail if one of the arguments of square roots in (4.3.7) or (4.3.8) is negative. This cannot happen for positive definite matrices, and the methods are sure to succeed even under roundoff so long as  $A$  is not too ill-conditioned. Notice that, as these square roots do not occur with the  $LDL^T$  factorization, such a factorization does not necessarily fail; for instance, if  $A$  is negative definite, it behaves just as if the  $L(-D)L^T$  factorization of  $-A$  were formed. However, it will fail if the leading diagonal entry of the Schur complement at any stage of the decomposition is zero and should be used with caution whenever  $A$  is indefinite. The method considered in the next section avoids this defect.

If  $A$  is sparse and positive definite, so is any symmetric permutation of  $A$ . Thus, provided  $A$  is not too badly conditioned, the Cholesky factorization of  $PAP^T$  will exist for any permutation  $P$ . Following on from Section 4.3.2, we may then be sure that any ordering chosen during a symbolic analysis of  $A$  will not be altered during the numerical factorization. The most common currently used heuristic for determining a good permutation for symmetric systems is the minimum-degree ordering. Quite simply, at the  $k$ th stage of the decomposition, the column with the fewest entries is symmetrically permuted to be the  $k$ th column of  $PAP^T$ . The aim of this strategy is to perform the least amount of work at each stage of the factorization. This does not mean, however, that the minimum-degree strategy results in the least possible work overall, or in the minimum possible fill-in. Other strategies, such as (locally) minimizing the fill-in, have also been proposed, but appear no better in general, with a higher cost.

### Notes and References for Subsection 4.3.3

The Cholesky and  $LDL^T$  factorizations of symmetric positive definite matrices are extremely stable. Wilkinson (1968) shows that so long as  $\epsilon_M c_n \kappa_2(A) \leq 1$ , then (4.3.4) is satisfied by  $\Delta b = 0$  and some  $\Delta A$  for which  $\|\Delta A\|_2 \leq \epsilon_M d_n \|A\|_2$ , where  $c_n$  and  $d_n$  are low-order polynomials in  $n$  and  $\epsilon_M$  is the relative machine precision (unit roundoff). More sophisticated componentwise bounds may be obtained. See Golub and Van Loan (1989, Section 5.2) and Higham (1996, Chapter 10) for details. Implementations are available within LAPACK (Anderson et al., 1995).

The minimum-degree and minimum-fill orderings are due to Tinney and Walker (1967). Other orderings, such as bandwidth reduction and nested dissection, are also possible, and preferable for certain special classes of matrices. See Duff, Erisman, and Reid (1986) and George and Liu (1981) for full details, and Duff (1997) for a survey and assessment of more recent developments. There are a number of codes for sparse problems, the best known being SPARSPAK (George and Liu, 1979), while performance issues are discussed by Dongarra et al. (1998).

---

<sup>30</sup>The  $(k + 1)$ st *pivot* is the leading diagonal entry in the Schur complement of  $A_{kk}$  in  $A$ .

#### 4.3.4 Symmetric Indefinite Matrices

When  $A$  is symmetric and indefinite, its  $LDL^T$  factorization may not exist.<sup>31</sup> In its place we have the symmetric indefinite decomposition

$$A = PLBL^TP^T, \quad (4.3.9)$$

where  $P$  is a permutation matrix,  $L$  is unit lower triangular, and  $B$  is block diagonal with each block being of dimension at most 2. We refer to the blocks as 1 by 1 and 2 by 2 pivots. Notice that the inertia of  $A$ —the numbers of positive, negative, and zero eigenvalues of  $A$ —is trivially obtained by summing the inertia of the pivots.

The permutation matrix  $P$  and the 2 by 2 pivots are both necessary to ensure that the decomposition exists and is stable. The permutations aim to avoid small 1 by 1 pivots, while the 2 by 2 pivots are present for the case in which no suitable 1 by 1 pivots are available. Unlike for the Cholesky or  $LDL^T$  factorizations, the permutations are required, and are applied, as the factorization proceeds. This has serious implications for sparse matrices, as now a good symbolic ordering may be ruined by such permutations. At each stage of the decomposition, specific technical rules are applied to determine whether a 1 by 1 pivot is more or less liable to lead to large entries in the resulting Schur complement than a 2 by 2 pivot; the pivot type that gives the lower a priori bound on the resulting entries is chosen. Typically, whenever  $A$  is definite, no 2 by 2 pivots occur.

#### Notes and References for Subsection 4.3.4

Such a factorization was first proposed by Bunch and Parlett (1971) and later improved by Bunch and Kaufman (1977) and Fletcher (1976) in the dense case and by Duff et al. (1979) and Duff and Reid (1983) in the sparse case. The decomposition, although potentially less stable than those for positive definite matrices, may be considered a reliable method for solving linear systems. In particular, the bound (4.3.4) typically is satisfied by  $\Delta b = 0$  and some  $\Delta A$  for which  $\|\Delta A\|_2 \leq \epsilon_M e_n g_n \|A\|_2$ , where  $e_n$  is a low-order polynomial in  $n$ , the *growth factor*  $g_n = ((1 + \sqrt{17})/8)^{n-1}$ , and, once again,  $\epsilon_M$  is the relative machine precision; the extra growth factor  $g_n$ , not present with the Cholesky or  $LDL^T$  factorizations, is a cause for concern in theory but is almost never seen in practice. However, recently, Ashcraft, Grimes, and Lewis (1995) and Higham (1995) have exposed a potentially serious flaw in the approach, which is that the norm of the generated factor  $L$  may be unbounded relative to  $\|A\|$ . As Higham (1995) has shown, this does not necessarily lead to instability; nevertheless a more restricted form of pivoting, as typified by the proposal of Ashcraft, Grimes, and Lewis (1995), may be required to ensure that  $\|L\|$  stays bounded. Interestingly, the sparse method proposed by Duff and Reid (1983) and implemented within the Harwell Subroutine Library (2000) code **MA27** already provided a suitably bounded  $\|L\|$ .

---

<sup>31</sup>For instance, there is no  $LDL^T$  factorization of any symmetric permutation of the indefinite matrix

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

### 4.3.5 Frontal and Multifrontal Methods

Matrices in optimization often arise in the form

$$A = \sum_{i=1}^e A_i,$$

where each of the component matrices  $A_i$  is symmetric and involves a small subset of the indices 1 to  $n$ . Such matrices are more commonly associated with finite-element methods for solving partial differential equations, and, by analogy, we shall refer to the  $A_i$  as *element* matrices, while the indices of  $A_i$  are its components.

It is perfectly possible to assemble (sum) the elements to form  $A$  and then to apply the methods considered in the last two sections. However, there is another class of methods that do not require that  $A$  be assembled when finding its factorization. These are known as *frontal* methods. The methods behave exactly as if  $A$  had been assembled (that is, a Cholesky,  $LDL^T$ , or symmetric indefinite factorization of  $A$  will result) but at each stage require only that those elements involving the current pivot(s) have been assembled. The other important ingredient is that all key computations take place on a dense matrix—the frontal matrix, or front for short—which comprises the rows and columns of all elements that have been assembled but that have not yet been used as pivots. The  $k$ th pivoting step thus requires firstly that all elements that involve the pivot but that have not yet been assembled now be introduced into the front—the frontal matrix might then expand; secondly, that the pivoting operations take place; and lastly, that the pivot column be removed from the front and the corresponding column of  $L$  recorded. It is easy to see that, because all the columns corresponding to entries in the pivot row lie in the front (the components must have been assembled), all of the operations (4.3.8) occur within the front. Slight complications occur when the matrix is indefinite, but these normally only mean that the frontal matrix grows larger than before while unsuitable potential pivots are left on hold.

The *multiprocedural* method is a generalization of this based on the observation that, at the start of the factorization of a sparse matrix, many pivots may be performed independently and at the same time. Each of these pivots then defines its own front, and elements are summed into individual fronts as required. At certain stages, it may be necessary to merge two or more fronts to produce a single larger front. The advantages of the multiprocedural method are that the frontal matrices are typically much smaller, and that there is a natural parallelism in the procedure. The only disadvantage is the complication that managing more than one front brings.

#### Notes and References for Subsection 4.3.5

The frontal method is due to Irons (1970), while the multiprocedural method was proposed by Duff and Reid (1983) and is implemented as **MA27** within the Harwell Subroutine Library (2000). Extensions to handle matrices of the form (4.1.1) when  $D = 0$  are given by Duff and Reid (1996) and are available as **MA47** in the Harwell Subroutine Library (2000). The use of

multifrontal methods within trust-region methods for unconstrained minimization has been considered by Conn et al. (1994).

### 4.3.6 Matrix Modification

Another issue that sometimes arises is that of modifying a matrix so that its inertia satisfies given conditions. The most common case is when we are given a symmetric matrix  $A$  and wish to add a “small” symmetric modification  $E$  to  $A$  so that the resulting matrix  $A+E$  is “safely” positive definite. Modified matrices may then be used to define search directions in linesearch methods for unconstrained minimization, convex models for trust-region methods (see Section 6.5), preconditioners for conjugate gradient and Lanczos methods (see Section 5.1.6), and appropriate norms to define trust regions (see Sections 6.7, 7.4, and 7.7). By “small”, we mean here that  $E$  should have a norm of the same order as  $A$  and, more importantly, that if  $A$  is itself “safely” positive definite,  $E$  should be zero. The term “safely” is meant to imply that the smallest eigenvalue of  $A+E$  is bounded (uniformly, when a sequence of matrices are involved) away from zero.

There are three important classes of such methods: modified Cholesky or  $LDL^T$  methods, modified spectral methods, and modified symmetric indefinite factorization methods. In the first, the perturbation  $E$  is a diagonal matrix and is computed as the Cholesky or  $LDL^T$  factorization proceeds. One such method works in two phases. In the first phase, the usual factorization is computed except that, before the pivoting operations (4.3.8) are applied to the whole of  $A$ , the diagonal entries that would result are checked to ensure that they are positive; the initial diagonal entries are checked to ensure this is so. So long as all diagonals remain positive, the usual Cholesky/ $LDL^T$  factorization results. However, if, at any stage, a current or potential diagonal entry is nonpositive, the factorization enters its second (modified) phase. In this phase, any pivot that is smaller than the sum of absolute values of the off-diagonal values in the pivot column is changed to be at least as large as this sum, and the pivoting operations are applied with this modified diagonal term. This method is equally applicable to sparse problems and can easily be incorporated into frontal or multifrontal schemes.

A second class of methods is based on the spectral decomposition (2.2.2) (p. 17). The idea here is simply to replace the eigenvalues  $\Lambda$  with another matrix whose eigenvalues are positive. One possibility is to replace all eigenvalues smaller than  $\delta > 0$  by  $\delta$ ; another is to replace  $\Lambda$  by  $|\Lambda|$ , which leads to a matrix of the form (2.2.3) (p. 17); and a third is to replace each eigenvalue  $\lambda$  by

$$\begin{cases} \lambda & \text{if } \lambda \geq \delta \text{ or} \\ -\lambda & \text{if } \lambda \leq -\delta \text{ or} \\ \delta & \text{otherwise.} \end{cases}$$

Of course, such methods are only appropriate for matrices for which the spectral decomposition is possible, which effectively limits them to small matrices. Furthermore, the cost of the spectral decomposition may be significantly higher than that of obtain-

ing a modified Cholesky factorization. On the other hand, the norm of the modification here may be significantly smaller than that for other alternatives.

The final class of methods aims at a compromise between the first two. The idea is simply to form the symmetric indefinite factorization (4.3.9) and to modify the entries of the diagonal block matrix  $B$ . The simplest possibility is to compute the spectral decomposition of each diagonal block (this is trivial since the blocks are of dimension at most 2) and to replace the eigenvalues by suitable positive values just as in the previous class of methods.

### Notes and References for Subsection 4.3.6

The first modified Cholesky factorization was due to Gill and Murray (1974), while that described here is a simplified version of that proposed by Schnabel and Eskow (1991, 1999) (see also Eskow and Schnabel, 1991). Both produce suitably bounded modifications, with the later bound being a priori smaller. The modified spectral decomposition idea is due to Greenstadt (1967). The modified symmetric indefinite factorization method was proposed by Gill, Murray, Ponceleón, and Saunders (1992) and was used in a variety of circumstances by Schlick (1993) and Conn, Gould, and Toint (1992b, Chapter 3). See Cheng and Higham (1998) for a rigorous justification.

The modification of one or more blocks of matrices of the general form (4.1.1) so that the resulting modified matrix has a prescribed inertia is still in its infancy. See Gould (1999b) and Higham and Cheng (1998) for details.

## 4.4 Least-Squares Problems and Projections

### 4.4.1 Least-Squares Problems

Given an  $m$  by  $n$  matrix  $A$  and a vector  $g$  with  $n$  components, a very commonly occurring problem is to find a vector  $y$  for which  $A^T y$  is closest to  $g$  in some norm. While there are applications for which other norms are appropriate, by far the most common case occurs when the  $\ell_2$  norm is used, and thus we aim to<sup>32</sup>

$$\underset{y \in \mathbb{R}^m}{\text{minimize}} \quad \|A^T y - g\|_2. \quad (4.4.1)$$

Since any minimizer of (4.4.1) equivalently solves

$$\underset{y \in \mathbb{R}^m}{\text{minimize}} \quad \|A^T y - g\|_2^2 \equiv \langle y, A A^T y \rangle - 2\langle y, A g \rangle + \langle g, g \rangle, \quad (4.4.2)$$

this is commonly known as the linear *least-squares* problem. The fact that  $\|\cdot\|_2^2$  is convex and differentiable immediately shows that any solution to (4.4.2) must satisfy the *normal equations*

$$A A^T y = A g \quad (4.4.3)$$

---

<sup>32</sup>We break with tradition here, as most authors consider the problem  $\min \|Ax - b\|_2$ . The problems that will arise in this book are, by and large, of the form (4.4.1) involving the transpose of  $A$ .

because of Theorem 3.2.1 (p. 38). Notice that  $AA^T$  is positive semidefinite. The normal equations are always consistent and will have a unique solution if and only if  $m \leq n$  and  $A$  is of full rank. If we define the residual  $r = g - A^T y$ , (4.4.3) gives that  $Ar = 0$ , from which we deduce that  $y$  and  $r$  together satisfy the *augmented* system

$$\begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r \\ y \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \quad (4.4.4)$$

There are a number of ways of solving the least-squares problem. If  $n \geq m$  and  $A$  is of full rank, then the most obvious are to solve either the normal equations (4.4.3) or the augmented system (4.4.4). The former might be solved via a Cholesky or  $LDL^T$  factorization of  $AA^T$ , while the latter necessarily requires a symmetric indefinite factorization of

$$\begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix}. \quad (4.4.5)$$

Another possibility is to form an  $LQ$  or a complete orthonormal factorization of  $A$ . The  $LQ$  factorization is simply a decomposition

$$A = LQ,$$

where  $Q$  is an  $n$  by  $n$  orthonormal matrix and  $L$  is  $m$  by  $n$  and lower triangular.<sup>33</sup> The complete orthonormal factorization takes this one stage further and decomposes  $A$  as

$$A = U \begin{pmatrix} L & 0 \\ 0 & 0 \end{pmatrix} Q, \quad (4.4.6)$$

where  $U$  is an  $m$  by  $m$  orthonormal matrix, while now  $L$  is  $\text{rank}(A)$  by  $\text{rank}(A)$  and lower triangular. As the  $LQ$  factorization is really appropriate only when  $A$  is of full rank, we concentrate on (4.4.6). In this case, as  $Q$  and  $U$  are orthonormal,

$$A^T y - g = Q^T \begin{pmatrix} L^T & 0 \\ 0 & 0 \end{pmatrix} U^T y - g = Q^T \begin{pmatrix} L^T w_1 - Q_1 g \\ -Q_2 g \end{pmatrix},$$

where

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = w = U^T y, \quad Q = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix},$$

and  $w_1$  and  $Q_1$  are, respectively, the first  $\text{rank}(A)$  components and rows of  $w$  and  $Q$ . Thus

$$\min_y \|A^T y - g\|_2^2 = \min_w \left\| \begin{pmatrix} L^T w_1 - Q_1 g \\ -Q_2 g \end{pmatrix} \right\|,$$

and, since  $L$  is nonsingular, the minimizing components  $w_1 = L^{-T} Q_1 g$ . As the required minimizer satisfies

$$y = U \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

---

<sup>33</sup>Lower triangular in the sense that all values to the right of its leading diagonal are zero.

and since  $w_2$  is arbitrary, it is conventional to pick  $w_2 = 0$ , which gives the so-called minimum-norm least-squares solution, for which  $\|y\|_2$  is as small as possible. Finally, as we saw in Section 2.2, if we can afford to obtain a singular value decomposition (2.2.4) (p. 17) of  $A$ , the minimum-norm solution of (4.4.2) may be computed as  $y = U\Sigma^{+T}V^Tg$ .

The complete orthonormal factorization of  $A$  may be obtained using a sequence of plane reflectors (or rotations). Starting with  $L_0 = A$  and  $Q_0 = I$ , at the  $(k+1)$ st stage we have  $L_k = AQ_k$ , where  $Q_k$  is a product of the first  $k$  plane reflectors and the values of the first  $k$  rows of  $L_k$  are zero to the right of their diagonals. The  $(k+1)$ st plane reflector is applied by postmultiplying  $L_k$  to remove the entries to the right of the diagonal in its  $(k+1)$ st row, while leaving the entries to the left of the diagonal unchanged. The resulting matrix is  $L_{k+1}$ , and the same reflector is applied to  $Q_k$  to produce  $Q_{k+1}$ . This process terminates when there are no further nonzeros to the right of the diagonal in  $L_\ell$  for some  $\ell$ ,<sup>34</sup> and this gives the  $LQ$  factorization of  $A$ . To finish the complete orthonormal decomposition, a further sequence of plane reflectors are applied to the left of  $L_\ell$  to remove any entries in rows  $\ell+1$  to  $n$ , starting by removing those in column  $\ell$  and moving to the left at each stage until those in column 1 are removed. The resulting product of reflectors gives  $U$ , and the nonzero part of  $L_\ell$  will have been transformed into  $L$ .

### Notes and References for Subsection 4.4.1

Strictly speaking, neither the normal equations nor augmented systems methods for linear least squares are numerically stable. A perturbation analysis (see, for instance, Björck, 1996, Section 1.4) shows that the condition number for the linear least-squares problem depends on

$$\kappa(A) \left( 1 + \kappa(A) \frac{\|r\|_2}{\|A\|_2 \|y\|_2} \right),$$

while a perturbation analysis of the normal equations method shows a dependence on  $\kappa(A)^2$  regardless of the size of the residual  $\|r\|_2$  (see Björck, 1996, Section 2.2). Thus the normal equations method is usually only trustworthy when  $\kappa(A)$  is moderate. A symmetric indefinite factorization of (4.4.5) may itself involve a factorization of the Schur complement  $-AA^T$  if the first  $n$  pivots are taken from the leading diagonal block of (4.4.5). Since the errors arising from the subsequent factorization of this Schur complement may then depend on  $\kappa(A)^2$  regardless of the size of the residual  $\|r\|_2$ , the augmented system method is also potentially unstable. This defect may be avoided by noting that (4.4.4) is equivalent to solving

$$\begin{pmatrix} \alpha I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \alpha^{-1}r \\ y \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}$$

for any nonzero  $\alpha$ . One can then choose  $\alpha$  so that pivots other than from the leading diagonal occur. However, an appropriate choice of  $\alpha$  is somewhat difficult. See Björck (1996, Section 2.2) for details.

The  $QR$  and complete orthonormal factorization and singular value decomposition methods are all extremely stable. As we have indicated, the main difficulty is in correctly assessing

---

<sup>34</sup>In theory,  $\ell$  should be  $\text{rank}(A)$ , but the presence of rounding may make it difficult to ascertain if very small entries to the right of the diagonal are really rounded zeros.

the rank of  $A$ , but satisfactory numerical procedures are available. These and many other issues are considered in detail by, among others, Golub and Van Loan (1989, Chapter 6), Higham (1996, Chapters 18 and 19), Lawson and Hanson (1974), and Björck (1996).

It is also common to solve the least-squares problem in a (weighted)  $H$  norm (see Section 2.3.1), in which  $H$  is positive definite. Usually  $H$  is diagonal, but more general  $H$  are sometimes used. In theory, this changes very little. For instance, the normal equations become

$$AHA^T y = AHg,$$

while the resulting augmented system is

$$\begin{pmatrix} H^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r \\ y \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}.$$

This problem will be considered in a more general context in the next section.

#### 4.4.2 Projections, Range-, and Null-Space Bases

Another important problem we shall encounter is that of finding the closest point to a given point  $x_0$ , in some appropriate norm, to the affine subspace

$$\mathcal{L} = \{x \mid Ax = b\}.$$

This is known as a least-distance problem, and the minimizer of such a problem is known as the *projection* of  $x_0$  onto  $\mathcal{L}$ ; we have already seen this for more general convex sets in Section 3.1.5. As before, while this problem may be posed in any norm, it most commonly occurs in the  $\ell_2$  norm, or a related  $H$  norm. If we consider the problem in the more general  $H$  norm, we thus wish to solve the (equivalent) problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|x_0 - x\|_H^2 \quad \text{subject to } Ax = b, \quad (4.4.7)$$

and, since this problem is differentiable and convex, it follows from Theorem 3.2.4 that the required value satisfies the augmented system

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} g \\ b \end{pmatrix} \quad (4.4.8)$$

for some Lagrange multipliers  $y$ , and where we have written  $g = Hx_0$ . We may solve (4.4.7) in a number of ways. Most obviously, we could solve (4.4.8) directly using a symmetric indefinite factorization. Alternatively, as  $H$  is (by assumption) nonsingular, we might isolate  $x$  in the first block of (4.4.8) and substitute this into the second. This results in what is commonly called the *range-space* approach to solving (4.4.8), in which  $y$  is first obtained by solving

$$AH^{-1}A^T y = b - AH^{-1}g$$

and  $x$  is subsequently recovered from the system

$$Hx = g + A^T y. \quad (4.4.9)$$

Finally, suppose that we can find full-rank matrices  $R$  and  $N$  whose columns span, respectively, the range- and null-spaces of  $A$ . Then, any vector  $x$  may be expressed as

$$x = Rx^R + Nx^N, \quad (4.4.10)$$

and, since  $AN = R^T N = 0$ , the range-space component  $x^R$  may formally be found as  $x^R = (AR)^+ b$  whenever  $Ax = b$  is consistent. On substituting (4.4.10) into (4.4.9), and premultiplying by  $N^T$ , we find that

$$N^T H N x^N = N^T g - N^T H R x^R,$$

from which we may recover the null-space component  $x^N$  provided that we factorize the positive definite matrix  $N^T H N$ . This is known as a *null-space* approach to solving (4.4.8). A closely related null-space method is to use any  $x^F$  for which  $Ax^F = b$ ; it need not lie solely in the range-space of  $A$ . For such an  $x^F$ , letting  $x = x^F + x^C$ , we see that  $Ax^C = 0$ , and thus that we must have  $x^C = Nx^N$ , and hence

$$x = x^F + Nx^N, \quad (4.4.11)$$

for some other  $x^N$ . Again substituting (4.4.11) into (4.4.9), and premultiplying by  $N^T$ , we deduce that

$$N^T H N x^N = N^T g - N^T H x^F. \quad (4.4.12)$$

As before, we can recover  $x^N$  using any factorization of the positive definite matrix  $N^T H N$ .

Probably the most reliable method for computing range- and null-space bases is to obtain the singular value decomposition (2.2.4) (p. 17); in this case,  $R$  simply consists of the columns of  $V$  corresponding to nonzero singular values, while  $N$  is made up of the remaining columns. Such a decomposition is usually too expensive to obtain on all but small matrices, but fortunately the cheaper complete orthonormal factorization (4.4.6) gives the same information; here  $R$  is made up from the first  $\text{rank}(A)$  rows of  $Q$ , while  $N$  comprises the remaining rows. Both methods give orthonormal bases for the two spaces. Of course, when  $A$  is of full rank and  $m \leq n$ , the rows of  $A$  itself give a basis for its range-space, and a basis for its null-space can be found as

$$N = P \begin{pmatrix} -(A^R)^{-1} A^N \\ I \end{pmatrix}, \quad \text{where } A = (A^R \ A^N) P^T \quad (4.4.13)$$

and the permutation matrix  $P$  is chosen so that the  $m$  by  $m$  submatrix  $A^R$  is nonsingular. Such a method does not generate orthonormal bases, and it is not automatically clear whether a particular subset of the columns of  $A$  gives a good null-space basis, but this freedom is particularly valuable when aiming to find a set of columns for which  $A^N$  has a sparse factorization, and thus for which  $(A^R)^{-1} A^N$  may be computed efficiently.

We should also mention here that (4.4.8) also defines the solution of the equality-constrained quadratic program

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle x, Hx \rangle - \langle x, g \rangle \quad \text{subject to } Ax = b, \quad (4.4.14)$$

and thus that the range- and null-space methods are equally applicable for (4.4.14). More significantly, while  $H$  needs to be positive definite for (4.4.7), all that is required for (4.4.14) to have a unique solution value (i.e., any critical point is indeed a global minimizer) is that  $N^T H N$  be positive (semi)definite.

### Notes and References for Subsection 4.4.2

Once again, Lawson and Hanson (1974) and Björck (1996) cover this topic in considerable detail. Algorithms for computing a sparse null-space basis are given by Coleman and Pothen (1986, 1987). The taxonomy of range- and null-space methods is due to Gill, Murray, and Wright (1981), where the solution of equality-constrained quadratic programming problems is discussed in detail. See also Fletcher (1987a).

A relevant question is whether it is possible to find an everywhere-continuous basis  $N(x)$  for the null-space of a continuous matrix  $A(x)$ . Perhaps surprisingly, Byrd and Schnabel (1986) show that the answer is “no” except in very special cases, although Coleman and Sorensen (1984) and Gill, Murray, Saunders, Stewart, and Wright (1985) show that locally continuous null-space base matrices may be found.

# Chapter 5

---

---

## Krylov Subspace Methods

---

---

### 5.1 The Conjugate Gradient Method

The final chapter in our introduction is considerably longer than its predecessor. This is quite deliberate. While the material here is, as before, background, it plays such a crucial role in building algorithms for large-scale optimization problems that we feel justified in covering it in some detail. Moreover, while the interactions between the conjugate gradient and Lanczos methods are well known in linear algebra circles, we suspect that the correspondence between the methods is not as well appreciated in the optimization community.

We have seen that the minimizer,  $x_*$ , of the quadratic minimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad q(x) = \frac{1}{2}\langle x, Hx \rangle + \langle c, x \rangle \quad (5.1.1)$$

satisfies the first-order optimality conditions

$$\nabla_x q(x_*) \stackrel{\text{def}}{=} g(x_*) = Hx_* + c = 0,$$

or equivalently the system of linear equations  $Hx_* = -c$ , provided  $H$  is positive definite. Thus the solution is easy to obtain so long as we are able to factorize  $H$ . However, when  $n$  is large, the factorization of  $H$  may be expensive or even impossible, and we must seek alternative methods for solving (5.1.1).

The factorization approach to finding  $x_*$  is known as a *direct* method, as a predetermined, finite amount of computation is required. By contrast, the methods we shall consider in this section are *iterative* methods as they seek to find a sequence of iterates  $x_k$  whose limit is  $x_*$  without explicitly inverting or factorizing  $H$ . A characteristic feature of iterative methods is that information about the inverse of  $H$  is (implicitly) built up by sampling all, or parts, of  $H$ . Many iterative methods have been proposed for solving systems of linear equations, and thus for solving (5.1.1). In this section, we shall concentrate on one such method, the method of conjugate gradients.

There are a number of reasons for being particularly drawn to conjugate gradients. Firstly, and most important, the method is, to date, the best general-purpose iterative

scheme for solving (5.1.1). Secondly, the algorithm is extremely simple to state and easy to implement. Thirdly, the method has a prescriptive convergence theory, and progress can be observed at each iteration. Fourthly, the iteration may be stopped early and yet the terminating point still be of use. And finally, as we shall see in later sections, modifications of the method are extremely useful even when  $H$  is not positive definite.

### 5.1.1 Minimization in a Subspace

Given a point  $x$  and a subspace  $\mathcal{S}$ , a *manifold* is the set of all points  $x + y$ , where  $y \in \mathcal{S}$ . Before we discuss the conjugate gradient method, we start by considering the problem of minimizing  $q(x)$  on a sequence of nested manifolds of increasing dimension. Once one of these manifolds fills  $R^n$ ; the solution to (5.1.1) will have been found. We have the following elementary result.

**Lemma 5.1.1** Let  $\mathcal{S}$  be a subspace of  $R^n$ , and suppose that the columns of the matrix  $S$  are a basis for  $\mathcal{S}$ . Furthermore, suppose that  $S^T H S$  is nonsingular. Then the critical point,  $x^s$ , of  $q$  given by (5.1.1) in the manifold  $x + Ss$  is given by

$$x^s = x - S (S^T H S)^{-1} S^T g(x) \quad (5.1.2)$$

and satisfies  $S^T g(x^s) = 0$ . If  $S^T H S$  is positive definite,  $x^s$  is the minimizer of  $q$  in the manifold.

**Proof.** Let  $q^s(s) = q(x + Ss)$ . Then

$$\nabla_s q^s(s) = S^T H(x + Ss) + S^T c = S^T H S s + S^T g(x) = S^T g(x + Ss).$$

If  $\nabla_s q^s(s) = S^T H S$  is nonsingular, the required critical point is  $x + Ss$ , where  $\nabla_s q^s(s) = 0$ . If  $\nabla_s q^s(s) = S^T H S$  is positive definite, this critical point must be the global minimizer of  $q$  on the manifold.  $\square$

Now consider a sequence of  $(k+1)$ -dimensional subspaces  $\mathcal{S}_k$ ,  $k = 0, 1, \dots, n-1$ , for which  $\mathcal{S}_0 \subset \mathcal{S}_1 \subset \dots \subset \mathcal{S}_{n-1} = R^n$ . We suppose that the columns of  $S_k = (s_0 \dots s_k)$  form a basis for  $\mathcal{S}_k$ . Then Lemma 5.1.1 shows that the minimizer of  $q$  in the manifold  $x_0 + S_k s$  is

$$x_{k+1} = x_0 - S_k (S_k^T H S_k)^{-1} S_k^T g_0,$$

where  $g_0 = g(x_0)$ . Furthermore, as the minimizer,  $x_k$ , of  $q$  in the manifold  $x_0 + S_{k-1} \bar{s}$  lies in the manifold  $x_0 + S_k s$ , the lemma also shows that

$$x_{k+1} = x_k - S_k (S_k^T H S_k)^{-1} S_k^T g_k,$$

where  $g_k = g(x_k)$ . If each  $x_j$  has been so obtained, Lemma 5.1.1 also shows that

$$S_k^T g_{k+1} = 0, \quad (5.1.3)$$

or more generally that

$$\langle s_i, g_j \rangle = 0 \text{ for all } i < j.$$

Hence, all components of  $S_k^T g_k$  but the last are zero, which gives that

$$x_{k+1} = x_k - \langle s_k, g_k \rangle S_k (S_k^T H S_k)^{-1} e_k. \quad (5.1.4)$$

### 5.1.2 Conjugate Directions

We next consider the class of *conjugate direction* methods, which are a particular instance of the methods considered in the previous section. Two vectors  $u$  and  $v$  are said to be  *$H$ -conjugate*<sup>35</sup> if

$$\langle u, Hv \rangle = 0.$$

To improve on the recurrence (5.1.4), we have the following result.

**Lemma 5.1.2** Suppose that each  $p_j$  is  $H$ -conjugate to its predecessors, and that  $P_k = (p_0, \dots, p_k)$ . Furthermore, suppose that  $P_k^T H P_k$  is positive definite. Then  $x_{k+1}$ , the minimizer of  $q$  in the manifold  $x_0 + P_k p$ , is given by

$$x_{k+1} = x_k + \sigma_k p_k, \text{ where} \quad (5.1.5)$$

$$\sigma_k = -\langle p_k, g_k \rangle / \langle p_k, H p_k \rangle \text{ and} \quad (5.1.6)$$

$$g_{k+1} = g_k + \sigma_k H p_k, \quad (5.1.7)$$

and  $g_{k+1}$  satisfies the relationship

$$\langle p_i, g_{k+1} \rangle = 0 \text{ for } i = 0, \dots, k. \quad (5.1.8)$$

**Proof.** The matrix  $P_k^T H P_k$  is diagonal because of the assumed  $H$ -conjugacy of the columns of  $p_k$ . Hence,

$$(P_k^T H P_k)^{-1} e_k = 1 / (\langle p_k, H p_k \rangle) e_k,$$

and thus (5.1.5) and (5.1.6) are immediate following (5.1.4). The identity (5.1.7) follows directly on multiplying (5.1.5) by  $H$ , while (5.1.8) simply reiterates (5.1.3).  $\square$

Observe that (5.1.5)–(5.1.7) is merely a simple linesearch method for  $q(x)$ . Given the current estimate of the solution  $x_k$  and the search direction  $p_k$ , (5.1.5) is a step  $\sigma_k$  along the search direction, where the step (5.1.6) locates the minimizer of  $q$  in this direction. The identity (5.1.7) is simply the resulting gradient of  $q$  following the step.

Lemma 5.1.2 is important as it shows that if we can build an  $H$ -conjugate basis for a nested set of subspaces of increasing dimension, and if  $H$  is positive definite in

---

<sup>35</sup>Most authors also require that  $H$  be positive definite. We prefer the more general definition.

these subspaces, we can find the minimizer in each of the resulting manifolds from an extremely simple recurrence. The remarkable fact is that we can easily build such an  $H$ -conjugate basis. The mechanism, as we shall shortly see, is the method of conjugate gradients.

### 5.1.3 Generating Conjugate Directions

We start with the following elementary result.

**Lemma 5.1.3** Let  $\{y_0, y_1, \dots, y_{k+1}\}$  be any set of vectors. Suppose we generate  $p_j$  with the recurrence

$$p_0 = y_0 \text{ and} \quad (5.1.9)$$

$$p_{j+1} = y_{j+1} + \sum_{i=0}^j \beta_{ji} p_i \text{ for } j = 0, 1, \dots, k, \quad (5.1.10)$$

for any set of weights  $\beta_{ji}$ . Then, if  $r_{k+1}$  is any vector for which  $\langle p_i, r_{k+1} \rangle = 0$  for  $i = 0, \dots, k$ , we have that

$$\langle y_i, r_{k+1} \rangle = 0 \text{ for } i = 0, 1, \dots, k. \quad (5.1.11)$$

**Proof.** It follows immediately from (5.1.9)–(5.1.10) that

$$\text{span}\{p_0, p_1, \dots, p_j\} = \mathcal{Y}_j \stackrel{\text{def}}{=} \text{span}\{y_0, y_1, \dots, y_j\} \text{ for } j = 0, 1, \dots, k+1.$$

The result then follows directly from the assumption that  $\langle p_i, r_{k+1} \rangle = 0$  for  $i = 0, 1, \dots, k$ .  $\square$

We shall suppose throughout this section that any generated product  $\langle p_j, H p_j \rangle \neq 0$  for  $p_j \neq 0$ . This is clearly the case when  $H$  is positive definite, but may also happen, if we are fortunate, with indefinite  $H$ .<sup>36</sup> We shall consider the general case, in detail, in Section 5.2.

We now consider how to generate a sequence of conjugate directions. Suppose that  $\{p_0, p_1, \dots, p_k\}$  are such a set, and that we wish to compute a new member  $p_{k+1}$ . Let  $y_{k+1}$  be any vector that is linearly independent of  $\{p_0, p_1, \dots, p_k\}$ , and consider the vector

$$p_{k+1} = y_{k+1} + \sum_{i=0}^k \beta_{ki} p_i. \quad (5.1.12)$$

Now suppose that we try to pick the  $\beta_{ki}$  in (5.1.12) to make  $p_{k+1}$   $H$ -conjugate to the members of  $\{p_0, p_1, \dots, p_k\}$ . Then (5.1.12) and the  $H$ -conjugacy of the  $p_i$  require that

$$0 = \langle p_{k+1}, H p_j \rangle$$

---

<sup>36</sup>However, this is a clear sign of potential numerical instability when  $H$  is not positive definite.

$$\begin{aligned}
&= \langle y_{k+1}, Hp_j \rangle + \sum_{i=0}^k \beta_{ki} \langle p_i, Hp_j \rangle \\
&= \langle y_{k+1}, Hp_j \rangle + \beta_{kj} \langle p_j, Hp_j \rangle \text{ for } j = 0, \dots, k,
\end{aligned}$$

that is, that

$$\beta_{kj} = -\frac{\langle y_{k+1}, Hp_j \rangle}{\langle p_j, Hp_j \rangle} \text{ for } j = 0, \dots, k. \quad (5.1.13)$$

Hence the vector

$$p_{k+1} = y_{k+1} - \sum_{i=0}^k \frac{\langle y_{k+1}, Hp_i \rangle}{\langle p_i, Hp_i \rangle} p_i$$

is  $H$ -conjugate<sup>37</sup> to the members of  $\{p_0, p_1, \dots, p_k\}$ . For a general  $y_{k+1}$ , this construction is increasingly expensive as it requires both storage for the vectors  $p_i$  and the calculation of the  $k+1$  products  $\langle y_{k+1}, Hp_i \rangle$  for each  $i = 0, \dots, k$ .

To improve on this, we digress for a moment. Suppose  $r_0$  is a given vector and that  $r_{i+1}$ ,  $i = 0, \dots, k$ , are generated from the recurrence

$$\alpha_i = -\langle p_i, r_i \rangle / \langle p_i, Hp_i \rangle, \text{ where} \quad (5.1.14)$$

$$r_{i+1} = r_i + \alpha_i Hp_i. \quad (5.1.15)$$

Then we have the following important result.

**Lemma 5.1.4** Suppose that  $\{p_0, p_1, \dots, p_k\}$  are  $H$ -conjugate and that  $r_{k+1}$  is generated from (5.1.14)–(5.1.15). Then

$$\langle p_i, r_{k+1} \rangle = 0 \text{ for } i = 0, \dots, k. \quad (5.1.16)$$

**Proof.** The proof is by induction. To start,

$$\langle p_0, r_1 \rangle = \langle p_0, r_0 \rangle - \langle p_0, r_0 \rangle \langle p_0, Hp_0 \rangle / \langle p_0, Hp_0 \rangle = 0.$$

Now suppose that  $\langle p_i, r_j \rangle = 0$  for  $i = 0, \dots, j-1$ . Then

$$\langle p_i, r_{j+1} \rangle = \langle p_i, r_j \rangle + \alpha_j \langle p_i, Hp_j \rangle = \langle p_i, r_j \rangle = 0 \text{ for } i = 0, \dots, j-1,$$

where  $\langle p_i, Hp_j \rangle = 0$  because of the  $H$ -conjugacy of the vectors  $\{p_0, p_1, \dots, p_j\}$ . In addition,

$$\langle p_j, r_{j+1} \rangle = \langle p_j, r_j \rangle - \langle p_j, r_j \rangle \langle p_j, Hp_j \rangle / \langle p_j, Hp_j \rangle = 0.$$

Thus

$$\langle p_i, r_{j+1} \rangle = 0 \text{ for } i = 0, \dots, j,$$

which completes the induction for  $j = 0, \dots, k$ .  $\square$

---

<sup>37</sup>This is nothing other than Gram–Schmidt orthogonalization in the  $H$  metric.

This lemma is important because it shows that the conclusion (5.1.8) in Lemma 5.1.2 is true for a *second* reason, independent of the fact that  $g_{k+1}$  is the gradient of  $q$  at the minimizer in the stated manifold, but as a result of the form of the recurrence (5.1.6)–(5.1.7). It is also important because the recurrence (5.1.14)–(5.1.15) then ensures that the vector  $r_{k+1}$  satisfies the requirement of Lemma 5.1.3.

If  $r_k$  is generated according to the recurrence (5.1.14) and (5.1.15), we may replace (5.1.13) by the equivalent

$$\beta_{kj} = \frac{\langle y_{k+1}, r_{j+1} - r_j \rangle}{\langle p_j, r_j \rangle} \quad \text{for } j = 0, \dots, k. \quad (5.1.17)$$

This alternative does not as yet appear to offer improvements in storage or speed of calculation. However, if we restrict the choice of  $y_{k+1}$ , things improve considerably, for we have the following result.

**Lemma 5.1.5** Suppose that  $\{p_0, p_1, \dots, p_k\}$  are a set of  $H$ -conjugate directions, that  $r_k$  is generated according to (5.1.14)–(5.1.15), and that  $y_{k+1}$  is chosen to be linearly independent from  $\{p_0, p_1, \dots, p_k\}$  such that

$$\langle y_{k+1}, r_i \rangle = 0 \quad \text{for } i = 0, \dots, k. \quad (5.1.18)$$

Then the vector

$$p_{k+1} = y_{k+1} + \beta_k p_k, \quad (5.1.19)$$

where

$$\beta_k = \frac{\langle y_{k+1}, r_{k+1} \rangle}{\langle p_k, r_k \rangle}, \quad (5.1.20)$$

is  $H$ -conjugate to  $\{p_0, p_1, \dots, p_k\}$ .

**Proof.** Using assumption (5.1.18), the formula (5.1.17) becomes

$$\beta_{kj} = 0 \quad \text{for } j = 0, \dots, k-1, \quad \text{and} \quad \beta_{kk} = \frac{\langle y_{k+1}, r_{k+1} \rangle}{\langle p_k, r_k \rangle};$$

that is, all but the last  $\beta_{kj}$  are zero. The result then follows from (5.1.12) on redefining  $\beta_k = \beta_{kk}$ .  $\square$

The added requirement that  $y_{k+1}$  be orthogonal to  $\{r_0, r_1, \dots, r_k\}$  thus has profound effects on both storage and computational effort, as the short formulae (5.1.19)–(5.1.20) testify. Remarkably, there is a readily available vector with this additional property, the vector<sup>38</sup>  $y_{k+1} = -r_{k+1}$ .

The key here is to compare the condition (5.1.11) from Lemma 5.1.3 with the required conjugacy condition (5.1.18) in Lemma 5.1.5. The first says that  $\langle y_i, r_{k+1} \rangle = 0$

---

<sup>38</sup>The minus sign here is not important, merely a convention.

for all  $i = 0, 1, \dots, k$ . The second requires that  $\langle y_{k+1}, r_i \rangle = 0$  for  $i = 0, \dots, k$ . Thus, picking  $y_i = -r_i$  in the first implies

$$\langle r_i, r_{k+1} \rangle = 0 \text{ for } i = 0, 1, \dots, k, \quad (5.1.21)$$

and thus the second must be true with this choice of  $y_i$ . The other condition required by Lemma 5.1.5 is trivial, as (5.1.16) shows that  $-r_{k+1}$  is orthogonal to the members of  $\text{span}\{p_0, p_1, \dots, p_k\}$ , and thus is certainly linearly independent from them.

Combining the recurrences (5.1.14)–(5.1.15) and (5.1.19)–(5.1.20) with the choice  $y_{k+1} = -r_{k+1}$ , we have a simple method for generating conjugate directions. A slight variation is possible by noting that

$$\langle r_{k+1}, p_{k+1} \rangle = -\langle r_{k+1}, r_{k+1} \rangle + \beta_k \langle r_{k+1}, p_k \rangle = -\langle r_{k+1}, r_{k+1} \rangle = -\|r_{k+1}\|_2^2 \quad (5.1.22)$$

from (5.1.16) and (5.1.19), and thus that the terms  $\langle p_k, r_k \rangle$  may be replaced by  $-\|r_k\|_2^2$  in both (5.1.14) and (5.1.20). The form (5.1.22) is often preferred. We summarize our findings in Algorithm 5.1.1.

**Algorithm 5.1.1: Generating conjugate directions**

Given  $r_0$ , set  $p_0 = -r_0$  and, for  $k = 0, 1, \dots$ , perform the iteration

$$\alpha_k = \|r_k\|_2^2 / \langle p_k, H p_k \rangle, \quad (5.1.23)$$

$$r_{k+1} = r_k + \alpha_k H p_k, \quad (5.1.24)$$

$$\beta_k = \|r_{k+1}\|_2^2 / \|r_k\|_2^2, \quad (5.1.25)$$

$$p_{k+1} = -r_{k+1} + \beta_k p_k. \quad (5.1.26)$$

Note that the dominant cost of each iteration is likely the matrix-vector product  $H p_k$ . One significant aspect is that no direct access to the elements of  $H$  is required, merely a means of forming the product with  $p_k$ . This is useful, for instance, when the action of  $H$  on a vector is available as a subroutine. A good example might be in automatic differentiation, where it is significantly less expensive to calculate  $H p_k$  than  $H$ .

We are free to choose whatever vector  $r_0$  we like, but the choice made completely determines the generated conjugate directions. As we have already seen, the spaces

$$\text{span}\{p_0, p_1, \dots, p_j\} \text{ and } \text{span}\{r_0, r_1, \dots, r_j\}$$

are the same. The relationship (5.1.24) shows that both spaces are identical to the *Krylov* (sub)space

$$\mathcal{K}(H, r_0, j) \stackrel{\text{def}}{=} \text{span}\{r_0, H r_0, H^2 r_0, \dots, H^j r_0\}.$$

Notice that the space is defined by three quantities: the matrix  $H$ , the initial vector  $r_0$ , and the highest power of  $H$ ,  $j$ . Thus, one interpretation of Algorithm 5.1.1 is that

it is the means of generating an  $H$ -conjugate basis for the Krylov space  $\mathcal{K}(H, r_0, j)$ . For the future, we record our findings that

$$\mathcal{K}(H, r_0, j) = \text{span} \{p_0, p_1, \dots, p_j\} = \text{span} \{r_0, r_1, \dots, r_j\}. \quad (5.1.27)$$

We shall shortly see a fourth representation of this Krylov space.

We make one final point before we consider the conjugate gradient method, which we state as Algorithm 5.1.2. It is often useful to be able to estimate the condition number of  $H$ , as this gives us some means of assessing the sensitivity of the solution to data perturbations (see Section 4.3.1). As we have seen, the Rayleigh quotient  $\langle v, Hv \rangle / \|v\|_2^2$  lies between the left- and rightmost eigenvalues of  $H$  for all nonzero  $v$ . As we compute  $\langle p_k, Hp_k \rangle$  during Algorithm 5.1.1, it would be satisfying to be able to compute  $\|p_k\|_2^2$  and thus the Rayleigh quotient at low cost. Fortunately, this is possible, as

$$\|p_{k+1}\|_2^2 = \|r_{k+1}\|_2^2 + \beta_k^2 \|p_k\|_2^2 \quad (5.1.28)$$

follows from (5.1.16) and (5.1.26). As we know,  $\|p_0\|_2 = \|r_0\|_2$ , and (5.1.28) provides a simple recurrence for  $\|p_{k+1}\|_2^2$  involving already computed quantities.

### 5.1.4 Conjugate Gradients

In the previous two sections, we have discovered that conjugate directions are useful as a means of generating approximations to the minimizer of  $q$  (Lemma 5.1.2), and that they may be generated extremely efficiently (Algorithm 5.1.1). In this section, we put the two aspects together to define the *conjugate gradient* method. In principle, the recurrences (5.1.5)–(5.1.7) and (5.1.23)–(5.1.26) may be performed independently. In particular, the vectors  $x_k$  and  $g_k$  are not required in Algorithm 5.1.1, nor is the vector  $r_k$  in Lemma 5.1.2. However, if we pick  $r_0 = g_0$ , the relationships (5.1.6)–(5.1.7) and (5.1.23)–(5.1.24) are the same and

$$r_j = g_j \quad \text{for all } j \geq 0. \quad (5.1.29)$$

This single additional observation defines the conjugate gradient method.

#### Algorithm 5.1.2: The conjugate gradient method

Given  $x_0$ , set  $g_0 = Hx_0 + c$  and let  $p_0 = -g_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\begin{aligned} \alpha_k &= \|g_k\|_2^2 / \langle p_k, Hp_k \rangle, \\ x_{k+1} &= x_k + \alpha_k p_k, \\ g_{k+1} &= g_k + \alpha_k H p_k, \\ \beta_k &= \|g_{k+1}\|_2^2 / \|g_k\|_2^2, \\ p_{k+1} &= -g_{k+1} + \beta_k p_k. \end{aligned}$$

Note that (5.1.27) and (5.1.29) imply that

$$\mathcal{K}(H, g_0, j) = \text{span} \{p_0, p_1, \dots, p_j\} = \text{span} \{g_0, g_1, \dots, g_j\}. \quad (5.1.30)$$

In Algorithm 5.1.2, we are implicitly assuming that  $H$  is positive definite over the generated Krylov space. A more realistic assumption is therefore that  $H$  itself be positive definite, particularly as we have little a priori control over the interaction between  $H$  and  $\mathcal{K}(H, g_0, j)$ , for  $j \geq 0$ .

### 5.1.5 Convergence of the Conjugate Gradient Method

We now consider the convergence behaviour of the conjugate gradient Algorithm 5.1.2 in *exact* arithmetic. For simplicity, we assume throughout this section that  $H$  is positive definite. We first observe that as  $x_{k+1}$  gives the minimizer of  $q(x)$  in the manifold  $x_0 + \mathcal{K}(H, g_0, k)$ , and as the dimension of the Krylov space  $\mathcal{K}(H, g_0, k)$  increases by 1 at each iteration, we must have  $x_k = x_*$  for some  $k \leq n$ ; that is, convergence must occur by the  $n$ th iteration. Of course, if  $g_k = 0$ , convergence will occur for some  $k < n$ . While this result tells us when we can (in theory) expect convergence, it does not help us predict the behaviour of the convergence. To do this, we need to consider the error in the solution at each stage of the conjugate gradient method.

We define the error in the solution as

$$\epsilon_k = x_k - x_*,$$

where, of course,

$$g(x_*) = Hx_* + c = 0.$$

It is immediate that

$$g_k = Hx_k + c = Hx_k - Hx_* = H\epsilon_k$$

and hence, following (5.1.30),

$$\mathcal{K}(H, g_0, k) = \text{span} \{H\epsilon_0, H^2\epsilon_0, \dots, H^{k+1}\epsilon_0\}. \quad (5.1.31)$$

A more revealing result follows from the identity

$$\begin{aligned} q(x) - q(x_*) &= \frac{1}{2}\langle x, Hx \rangle + \langle c, x \rangle - \frac{1}{2}\langle x_*, Hx_* \rangle - \langle c, x_* \rangle \\ &= \frac{1}{2}\langle x, Hx \rangle - \frac{1}{2}\langle x_*, Hx_* \rangle + \langle x - x_*, c \rangle \\ &= \frac{1}{2}\langle x, Hx \rangle - \frac{1}{2}\langle x_*, Hx_* \rangle + \langle x_* - x, Hx_* \rangle \\ &= \frac{1}{2}\langle x - x_*, H(x - x_*) \rangle = \frac{1}{2}\|x - x_*\|_H^2, \end{aligned}$$

where the  $H$  norm is as defined by (2.3.2) (p. 22) in Section 2.3. We recall from Lemma 5.1.1 that  $x_k$  gives the minimizer of  $q(x)$  in the manifold  $x_0 + \mathcal{K}(H, g_0, k-1)$ . Hence

$$\begin{aligned} 2 \min_{x \in x_0 + \mathcal{K}(H, g_0, k-1)} (q(x) - q(x_*)) &= 2(q(x_k) - q(x_*)) \\ &= \|\epsilon_k\|_H^2 \\ &= \min_{x \in x_0 + \mathcal{K}(H, g_0, k-1)} \|x - x_*\|_H^2 \\ &= \min_{\epsilon \in \epsilon_0 + \mathcal{K}(H, g_0, k-1)} \|\epsilon\|_H^2. \end{aligned} \quad (5.1.32)$$

Notice that (5.1.32) implies that the error (measured in the  $H$  norm) decreases at every iteration. Next, we examine the manifold  $\epsilon_0 + \mathcal{K}(H, g_0, k - 1)$ . The identity (5.1.31) reveals that any vector  $\epsilon$  in this manifold must be of the form

$$\epsilon = \epsilon_0 + \sum_{i=1}^k \gamma_i H^i \epsilon_0 = \phi_k(H) \epsilon_0,$$

where  $\phi_k$  is a polynomial of degree  $k$  whose constant term is 1, i.e.,  $\phi_k(0) = 1$ ; we denote the set of all such polynomials as

$$\mathcal{P}_k = \{\text{polynomials } \phi \text{ of degree } k \text{ for which } \phi(0) = 1\}. \quad (5.1.33)$$

Combining (5.1.32)–(5.1.33) we thus have

$$\|\epsilon_k\|_H^2 = \min_{\phi \in \mathcal{P}_k} \|\phi(H) \epsilon_0\|_H^2. \quad (5.1.34)$$

As  $H$  is symmetric, we may use its spectral decomposition to simplify matters. Suppose that  $\Lambda = \{\lambda_1, \dots, \lambda_n\}$  are eigenvalues of  $H$  and  $u_1, \dots, u_n$  are the corresponding orthonormal eigenvectors. Then, on expressing  $\epsilon_0$  as a linear combination

$$\epsilon_0 = \sum_{i=1}^n \theta_i u_i$$

of these eigenvectors, we deduce that

$$\begin{aligned} \|\phi(H) \epsilon_0\|_H^2 &= \left\langle \sum_{j=1}^n \phi(H) \theta_j u_j, \sum_{i=1}^n H \phi(H) \theta_i u_i \right\rangle \\ &= \left\langle \sum_{j=1}^n \theta_j \phi(\lambda_j) u_j, \sum_{i=1}^n \theta_i \lambda_i \phi(\lambda_i) u_i \right\rangle \\ &= \sum_{i=1}^n \theta_i^2 \lambda_i \phi(\lambda_i)^2 \\ &\leq \max_{\lambda \in \Lambda} \phi(\lambda)^2 \sum_{i=1}^n \theta_i^2 \lambda_i \\ &= \max_{\lambda \in \Lambda} \phi(\lambda)^2 \|\epsilon_0\|_H^2. \end{aligned} \quad (5.1.35)$$

Combining (5.1.34) and (5.1.35), we deduce that

$$\frac{\|\epsilon_k\|_H}{\|\epsilon_0\|_H} \leq \min_{\phi \in \mathcal{P}_k} \max_{\lambda \in \Lambda} |\phi(\lambda)|, \quad (5.1.36)$$

this inequality being an equation if  $\epsilon_0$  lies along an appropriate eigenvector of  $H$ . We can deduce an immediate consequence of (5.1.36).

**Theorem 5.1.6** If  $H$  has  $\ell$  distinct eigenvalues, the conjugate gradient Algorithm 5.1.2 will terminate with  $x_j = x_*$  for some  $j \leq \ell$ .

**Proof.** The result follows immediately from (5.1.36). For if  $\Lambda$  only contains  $\ell$  distinct values  $\{\lambda_1, \dots, \lambda_\ell\}$ , it is possible to find a polynomial

$$p_\ell(\lambda) = \left(1 - \frac{\lambda}{\lambda_1}\right) \cdots \left(1 - \frac{\lambda}{\lambda_\ell}\right) \quad (5.1.37)$$

of degree  $\ell$  for which  $p_\ell(0) = 1$  and  $p_\ell(\lambda_i) = 0$  for  $i = 1, \dots, \ell$ . Hence

$$\min_{\phi \in \mathcal{P}_\ell} \max_{\lambda \in \Lambda} |\phi(\lambda)| = 0,$$

and therefore  $\epsilon_\ell = 0$ .  $\square$

A number of error bounds may be deduced from (5.1.36) by broadening the admissible set of  $\lambda$ . In particular, (5.1.36) implies

$$\frac{\|\epsilon_k\|_H}{\|\epsilon_0\|_H} \leq \min_{\phi \in \mathcal{P}_k} \max_{\lambda \in \mathcal{L}} |\phi(\lambda)|, \quad (5.1.38)$$

where  $\Lambda \subseteq \mathcal{L}$ . The best-known result is given by the following theorem.

**Theorem 5.1.7** The error  $\epsilon_k = x_k - x_*$  of the iterates generated by the conjugate gradient Algorithm 5.1.2 satisfies the inequality

$$\frac{\|\epsilon_k\|_H}{\|\epsilon_0\|_H} \leq 2 \left( \frac{\sqrt{\kappa(H)} - 1}{\sqrt{\kappa(H)} + 1} \right)^k, \quad (5.1.39)$$

where  $\kappa(H)$  is the spectral condition number of  $H$ .

**Proof.** Pick the set

$$\mathcal{L} = \{\lambda \in [\lambda_{\min}, \lambda_{\max}]\}, \quad (5.1.40)$$

where  $\lambda_{\min}$  and  $\lambda_{\max}$  are the smallest and largest eigenvalues of  $H$ . In this case, the value on the right of inequality (5.1.38) is the solution to a classical problem from approximation theory: of all the polynomials of degree  $k$  for which  $p(0) = 1$ , find the one that has smallest absolute maximum value over the interval  $[\lambda_{\min}, \lambda_{\max}]$ . The solution to this problem is the (shifted and weighted) Chebyshev polynomial

$$T_k \left( \frac{2\lambda - \lambda_{\max} - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right) / T_k \left( -\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right),$$

where

$$T_k(x) = \cos(k \arccos x) \equiv \frac{1}{2} \left( x + \sqrt{x^2 - 1} \right)^k + \frac{1}{2} \left( x - \sqrt{x^2 - 1} \right)^k,$$

which has the extremal absolute value

$$\frac{1}{T_k \left( \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right)} = \frac{1}{T_k \left( \frac{\kappa(H) + 1}{\kappa(H) - 1} \right)} \quad (5.1.41)$$

(see Powell, 1981a, Section 7.3). Simple manipulation then shows that

$$T_k \left( \frac{\kappa(H) + 1}{\kappa(H) - 1} \right) = \frac{1}{2} \left( \frac{\sqrt{\kappa(H)} + 1}{\sqrt{\kappa(H)} - 1} \right)^k + \frac{1}{2} \left( \frac{\sqrt{\kappa(H)} - 1}{\sqrt{\kappa(H)} + 1} \right)^k \geq \frac{1}{2} \left( \frac{\sqrt{\kappa(H)} + 1}{\sqrt{\kappa(H)} - 1} \right)^k. \quad (5.1.42)$$

Finally, using (5.1.38) for the particular  $\mathcal{L}$  in (5.1.40) with the bound (5.1.42) on the value (5.1.41) gives the required result.  $\square$

The basic message from Theorem 5.1.7 is that the worse the conditioning of  $H$ , the slower the likely convergence of the conjugate gradient method. Of course, a number of inequalities were used in the derivation of this result and thus it may happen that (5.1.39) is pessimistic. However, considerable practical experience has indicated that the bound is actually quite realistic unless  $H$  really has special properties.

Tighter results are possible if more information on the spectrum of  $H$  is available. For instance, if  $\Lambda \subseteq [a, b] \cup [c, d]$  for some known values  $a, b, c$ , and  $d$ , we merely need to ensure that  $\phi$  is small in the intervals  $[a, b]$  and  $[c, d]$  and need not be concerned with its behaviour outside these intervals.

Another consequence of (5.1.36) is that if the eigenvalues of  $H$  are clustered around  $\ell$  distinct values, the relative error will be small after at most  $\ell$  iterations. For, if  $\{\lambda_1, \dots, \lambda_\ell\}$  are these distinct values, the polynomial  $p_\ell(\lambda)$  in (5.1.37) is zero at these values and satisfies  $p_\ell(0) = 1$ . By continuity  $p_\ell(\lambda)$  will be small at the remaining eigenvalues, which are clustered around  $\{\lambda_1, \dots, \lambda_\ell\}$ ; thus

$$\min_{\phi \in \mathcal{P}_\ell} \max_{\lambda \in \Lambda} |\phi(\lambda)|,$$

and hence  $\|\epsilon_\ell\|_H / \|\epsilon_0\|_H$ , will be small.

### 5.1.6 Preconditioned Conjugate Gradients

We have seen from Theorem 5.1.7 that the convergence behaviour of the conjugate gradient method is strongly dependent on the conditioning of  $H$ , and the larger the condition number, the slower the likely convergence of the method. From the proof of Theorem 5.1.6 we see that the more the eigenvalues are clustered, the sooner large reductions in the error will occur. The aim of *preconditioning* is to transform the problem to improve the convergence of the conjugate gradient method. Quite simply, we aim to solve the *transformed* or *preconditioned* problem

$$\underset{\bar{x} \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle \bar{x}, \bar{H} \bar{x} \rangle + \langle \bar{c}, \bar{x} \rangle, \quad (5.1.43)$$

where

$$\bar{H} = R^{-T} H R^{-1} \quad \text{and} \quad \bar{c} = R^{-T} c, \quad (5.1.44)$$

and to recover

$$x_* = R^{-1} \bar{x}_*. \quad (5.1.45)$$

The intention is that the invertible transformation matrix  $R$  be chosen both to cluster and to reduce the spread of the eigenvalues of the transformed Hessian  $R^{-T}HR^{-1}$ . However, we do not want to explicitly form  $\bar{H}$  and  $\bar{c}$ , since  $\bar{H}$  might be dense while  $H$  is sparse. We would rather derive an implicit method based on the raw data  $H$  and  $c$ . This is easy to achieve. What is difficult is to find an effective preconditioner.

To derive a suitable method, we apply Algorithm 5.1.2 to (5.1.43) and then transform the iteration back to the original variables. Applying Algorithm 5.1.2 to (5.1.43) leads to the following algorithm.

**Algorithm 5.1.3: Conjugate gradient method for the transformed problem**

Given  $\bar{x}_0$ , set  $\bar{g}_0 = \bar{H}\bar{x}_0 + \bar{c}$  and let  $\bar{p}_0 = -\bar{g}_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\alpha_k = \|\bar{g}_k\|_2^2 / \langle \bar{p}_k, \bar{H}\bar{p}_k \rangle, \quad (5.1.46)$$

$$\bar{x}_{k+1} = \bar{x}_k + \alpha_k \bar{p}_k, \quad (5.1.47)$$

$$\bar{g}_{k+1} = \bar{g}_k + \alpha_k \bar{H}\bar{p}_k, \quad (5.1.48)$$

$$\beta_k = \|\bar{g}_{k+1}\|_2^2 / \|\bar{g}_k\|_2^2, \quad (5.1.49)$$

$$\bar{p}_{k+1} = -\bar{g}_{k+1} + \beta_k \bar{p}_k. \quad (5.1.50)$$

Now, define

$$\bar{x}_k = Rx_k, \quad \bar{p}_k = Rp_k, \quad \text{and} \quad \bar{g}_k = R^{-T}g_k. \quad (5.1.51)$$

If we substitute (5.1.51) into (5.1.47) and (5.1.50) and premultiply by  $R^{-1}$ , we immediately have

$$x_{k+1} = x_k + \alpha_k p_k \quad \text{and} \quad p_{k+1} = -R^{-1}R^{-T}g_{k+1} + \beta_k p_k. \quad (5.1.52)$$

Likewise, substituting (5.1.51) into (5.1.48), multiplying the result by  $R^T$ , and using (5.1.44) yields

$$g_{k+1} = g_k + \alpha_k H p_k.$$

Finally, as  $\langle \bar{p}_k, \bar{H}\bar{p}_k \rangle = \langle p_k, H p_k \rangle$  and

$$\|\bar{g}_k\|_2^2 = \langle R^{-T}g_k, R^{-T}g_k \rangle = \langle g_k, R^{-1}R^{-T}g_k \rangle,$$

we may rewrite (5.1.46) and (5.1.49) as

$$\alpha_k = \frac{\langle g_k, R^{-1}R^{-T}g_k \rangle}{\langle p_k, H p_k \rangle} \quad \text{and} \quad \beta_k = \frac{\langle g_{k+1}, R^{-1}R^{-T}g_{k+1} \rangle}{\langle g_k, R^{-1}R^{-T}g_k \rangle}.$$

Thus, if we define the *preconditioner*

$$M = R^T R, \quad (5.1.53)$$

and let

$$v_k = M^{-1}g_k \equiv R^{-1}R^{-T}g_k, \quad (5.1.54)$$

the relationships (5.1.52)–(5.1.54) define Algorithm 5.1.4, the *preconditioned conjugate gradient method*.

**Algorithm 5.1.4: The preconditioned conjugate gradient method**

Given  $x_0$ , set  $g_0 = Hx_0 + c$ , and let  $v_0 = M^{-1}g_0$  and  $p_0 = -v_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\alpha_k = \langle g_k, v_k \rangle / \langle p_k, H p_k \rangle, \quad (5.1.55)$$

$$x_{k+1} = x_k + \alpha_k p_k,$$

$$g_{k+1} = g_k + \alpha_k H p_k,$$

$$v_{k+1} = M^{-1}g_{k+1}, \quad (5.1.56)$$

$$\beta_k = \langle g_{k+1}, v_{k+1} \rangle / \langle g_k, v_k \rangle, \quad (5.1.57)$$

$$p_{k+1} = -v_{k+1} + \beta_k p_k. \quad (5.1.58)$$

It is important to notice that the eigenvalues of  $M^{-1}H$  are the same as those of  $\bar{H} \equiv R^{-T}HR^{-1}$  (the two matrices are similar—see Section 2.2), and therefore any attempt to influence the eigenvalues of  $\bar{H}$  may be achieved directly using  $M$ . Thus, although we have motivated the preconditioned conjugate gradient method via the transformation  $R$ , it is more convenient to define the method in terms of  $M$ , and there is no actual need to find  $R$ . Notice, however, that it is implicit in (5.1.53) and the assumption that  $R$  is nonsingular that  $M$  is symmetric and positive definite.

The Krylov space investigated by the preconditioned algorithm is

$$\mathcal{K}(M^{-1}H, M^{-1}g_0, j),$$

while (5.1.16) becomes

$$\langle p_i, Mv_{k+1} \rangle = 0 \text{ for } i = 0, \dots, k. \quad (5.1.59)$$

There is no longer a simple recurrence to obtain the Rayleigh quotient. However, (5.1.28) for the transformed problem is

$$\|\bar{p}_{k+1}\|_2^2 = \|\bar{g}_{k+1}\|_2^2 + \beta_k^2 \|\bar{p}_k\|_2^2,$$

which becomes

$$\langle p_{k+1}, Mp_{k+1} \rangle = \langle g_{k+1}, v_{k+1} \rangle + \beta_k^2 \langle p_k, Mp_k \rangle \quad (5.1.60)$$

using (5.1.51) and (5.1.53). Thus the generalized Rayleigh quotient

$$\rho_{k+1} = \frac{\langle p_{k+1}, H p_{k+1} \rangle}{\langle p_{k+1}, Mp_{k+1} \rangle} \quad (5.1.61)$$

is essentially available for free from repeated application of (5.1.61). Comparing Algorithm 5.1.2 with Algorithm 5.1.4, it is apparent that the extra costs involved with the preconditioned variant are in finding and perhaps factorizing  $M$  and in forming (5.1.56) or, equivalently, in solving the linear system

$$Mv_{k+1} = g_{k+1}.$$

In its favour, we have two improved convergence results.

**Corollary 5.1.8** The error  $\epsilon_k = x_k - x_*$  of the iterates generated by the preconditioned conjugate gradient Algorithm 5.1.4 satisfies the inequality

$$\frac{\|\epsilon_k\|_H}{\|\epsilon_0\|_H} \leq 2 \left( \frac{\sqrt{\kappa(M^{-1}H)} - 1}{\sqrt{\kappa(M^{-1}H)} + 1} \right)^k, \quad (5.1.62)$$

where  $\kappa(M^{-1}H)$  is the spectral condition number of  $M^{-1}H$ .

**Proof.** Applying Theorem 5.1.7 to Algorithm 5.1.3 gives the bound

$$\frac{\|\bar{\epsilon}_k\|_{\bar{H}}}{\|\bar{\epsilon}_0\|_{\bar{H}}} \leq 2 \left( \frac{\sqrt{\kappa(\bar{H})} - 1}{\sqrt{\kappa(\bar{H})} + 1} \right)^k, \quad (5.1.63)$$

where  $\bar{\epsilon}_k = \bar{x}_k - \bar{x}_*$ . The relationships (5.1.44), (5.1.45), and (5.1.51) show that

$$\|\bar{\epsilon}_k\|_{\bar{H}}^2 = \langle \bar{\epsilon}_k, \bar{H}\bar{\epsilon}_k \rangle = \langle \epsilon_k, H\epsilon_k \rangle = \|\epsilon_k\|_H^2. \quad (5.1.64)$$

As Algorithm 5.1.3 is simply a transformed version of Algorithm 5.1.4, (5.1.63), and (5.1.64), and the similarity of  $\bar{H}$  and  $M^{-1}H$  then lead directly to (5.1.62).  $\square$

**Corollary 5.1.9** If  $M^{-1}H$  has  $\ell$  distinct eigenvalues, the preconditioned conjugate gradient Algorithm 5.1.4 will terminate with  $x_j = x_*$  for some  $j \leq \ell$ .

**Proof.** The result follows directly by applying Theorem 5.1.6 to the iterates generated by Algorithm 5.1.3 and by the noted equivalence of the transformed iterates generated by this algorithm and Algorithm 5.1.4.  $\square$

Thus, it is now the matrix  $M^{-1}H$  that determines the convergence behaviour of the method. Corollary 5.1.8 shows that the worst-case convergence rate depends upon the condition number of  $M^{-1}H$ , and thus that we should aim to choose  $M$  to make this number as small as possible. Corollary 5.1.9 indicates that we should also aim to

pick  $M$  to cluster the eigenvalues of  $M^{-1}H$  around, say,  $\ell$  values. The relative error will then be small after at most  $\ell$  iterations.

In summary, an iteration of the preconditioned conjugate gradient method is more expensive than its unpreconditioned sister, but a judicious choice of preconditioner may reduce the number of iterations required to achieve a given level of accuracy in the solution. Thus, the overall cost of the preconditioned method may actually be smaller. The difficulty is, of course, in finding a preconditioner with the required spectral properties that is also inexpensive to use.

The two extremes<sup>39</sup> are the choices  $M = I$  and  $M = H$ . The former clearly does not change the spectrum of  $M^{-1}H$  at all, but is very inexpensive—the preconditioned conjugate gradient method is the ordinary method in this case. The latter choice essentially requires the inverse, or a full factorization, of  $H$ , and leads to convergence in a single iteration as  $M^{-1}H = I$ —the method is then a direct method. In between these extremes lie a vast number of possibilities. Many of the proposals that have been made are for special classes of matrices—for instance, matrices arising from the discretization of elliptic partial differential equations—and it appears to be a sad fact of life that devising an effective and efficient general-purpose preconditioner is impossible.

It is beyond the scope of this book to consider preconditioners in any detail, but we should at least mention some common suggestions. The simplest is to choose  $M$  as the diagonal of  $H$ .<sup>40</sup> This typically works well when  $H$  is very diagonally dominant,<sup>41</sup> as then the condition number of  $M^{-1}H$  is bounded by  $(1 + \delta)/(1 - \delta)$  for some  $\delta$  significantly smaller than 1. Clearly such a preconditioner is cheap to apply. A generalization is to pick  $M$  as a banded portion of  $H$ , discarding all nonzero  $h_{i,j}$  for which  $|i - j| > n_b$  for some preselected integer  $n_b$ . So long as  $n_b$  is small, obtaining a band factorization of  $M$  is inexpensive, and thus operations with the preconditioner are practical. Another important class of preconditioners is based on an incomplete Cholesky factorization of  $H$ . In its most basic form, any fill-in during the sparse Cholesky decomposition is discarded. Improvements are possible by allowing levels of fill-in, a level-1 method allowing fill-in from nonzeros in the original matrix, but discarding any fill-ins arising from created nonzeros. These methods are only guaranteed to succeed for certain classes of  $H$ , but may be modified to handle other cases. Another possibility is to approximate the inverse  $M^{-1}$  directly by minimizing  $\|M^{-1}H - I\|$  while requiring that  $M^{-1}$  be sparse. Finally, if  $H$  is known to have certain structural properties, a preconditioner may take this into account. For instance, if  $H$  has the form  $\sum_i H_i$  where the  $H_i$  are of low rank, element by element preconditioners that respect this structure may be constructed. As another example, if  $H$  has the form  $S + L$ , where  $S$  and  $L$  contain small and large components, and if  $L$  or  $L + E$  is easily invertible for some small perturbation  $E$ , preconditioners of the form  $M = L$  or  $M = L + E$  may prove

<sup>39</sup>One could actually imagine more extreme cases in which the conditioning or the clustering of the eigenvalues worsen.

<sup>40</sup>This is sometimes called the Jacobi preconditioner.

<sup>41</sup>A positive definite matrix is very diagonally dominant when each diagonal is much larger than the sum of the absolute values of the off-diagonals in its column.

effective, as the resulting  $M^{-1}H$  will have many eigenvalues clustered about 1.

## Notes and References for Section 5.1

The conjugate gradient method is due to Hestenes and Stiefel (1952). Good summaries are provided by Gill, Murray, and Wright (1981, Section 4.8.3) and Fletcher (1987a, Section 4.1). Conjugate direction methods are discussed by Luenberger (1969, Section 10.6), Hestenes (1980, Chapter 5), and Hackbusch (1994, Section 9.3). An annotated history of conjugate gradient methods up until 1976 is given by Golub and O’Leary (1989).

A convergence analysis of the method (Section 5.1.5) was first given by Kaniel (1966) and Daniel (1967a, 1967b). A good summary is provided by Stoer (1983, Section 1), while possible improvements are summarized by Axelsson (1996, Chapter 13). To paraphrase Axelsson (1996, Section 13.5), the true behaviour may be conveniently viewed as occurring in three phases. In the initial phase, the error  $\|\epsilon_k\|_H/\|\epsilon_0\|_H$  decreases at a rate that is independent of the condition number. The middle phase is characterized by the linear convergence predicted by Corollary 5.1.8. In the final phase, the convergence may accelerate to a superlinear rate if the condition number  $\kappa(M^{-1}H)$  is modest, or if the eigenvalues of  $M^{-1}H$  are sufficiently isolated from one another.

We have concentrated on the theoretical convergence behaviour of the method, but it would be foolish to pretend that its floating-point behaviour is identical. Indeed, while the method was originally proposed as a direct method for solving linear systems, early numerical experiments revealed that the expected worst-case  $n$ -step convergence did not always occur in practice, and it soon fell from favour. However, the method underwent a renaissance following Reid’s (1971) observation that it was better viewed as an iterative method from which a good estimate of the solution might be extracted. Nowadays, it is typically truncated; that is,  $x_k$  is accepted as an estimate of the solution when the norm of the gradient (or some other measure) has been reduced by a predefined amount. In practice, for many applications, a truncated estimate is often just as useful as the exact solution, while the savings in computation may be considerable.

In floating-point arithmetic, as has been shown by Strakoš (1991), Greenbaum and Strakoš (1992), and Notay (1993), the true convergence behaviour is as if the method in exact arithmetic were applied to a larger system. The three phases of convergence described by Axelsson (1996, Section 13.5) that we mentioned above are frequently observed in practice.

The idea of preconditioning the method has been proposed by a number of authors. See, for instance, Evans (1968), Axelsson (1972), Concus, Golub, and O’Leary (1976), and Lin and Moré (1999b). Good general discussions of preconditioners, including those we have mentioned in this section, are contained in the books by Axelsson (1996), Greenbaum (1997), and Saad (1996).

## 5.2 The Lanczos Method

Algorithm 5.1.1 gives us one way of generating conjugate directions. The cautious reader may rightly be concerned that these recurrences break down if the denominator

$\langle p_k, Hp_k \rangle$  in (5.1.23) is zero, and may prove unstable when  $\langle p_k, Hp_k \rangle$  is small.<sup>42</sup> Of course, this possibility cannot arise when  $H$  is sufficiently positive definite, but in many of the trust-region applications we shall encounter later in this book, indefinite  $H$  will occur. Thus we are interested in finding other means of generating conjugate directions.

The conjugate directions generated by Algorithm 5.1.1 form an  $H$ -conjugate basis for the Krylov space  $\mathcal{K}(H, r_0, j)$ . Let us examine this space in more detail. We suppose that

$$H = U\Lambda U^T$$

is the eigendecomposition (2.2.2) (p. 17) of  $H$ . Then it immediately follows from

$$H^j = (U\Lambda U^T)^j = U\Lambda^j U^T$$

that

$$\mathcal{K}(H, r_0, j) = U\mathcal{K}(\Lambda, U^T r_0, j).$$

Thus we see that the Krylov space will expand to fill  $\mathbb{R}^n$  unless one or more components of  $U^T r_0$  is zero, or if one or more of the eigenvalues of  $H$  is zero (or both). We shall say that the Krylov space *degenerates* if either of these possibilities occurs. Notice that the first cause of degeneracy is a consequence of an unlucky choice of  $r_0$ , while the second means that  $H$  is singular. A breakdown in Algorithm 5.1.1 is not a consequence of the degeneracy of  $\mathcal{K}(H, r_0, j)$  (although the algorithm will certainly break down if the space is degenerate), but a result of a poor choice of basis.

The method we shall now consider, the Lanczos method, aims to form an *orthonormal* basis for  $\mathcal{K}(H, r_0, j)$ , and thus will only fail when the space itself degenerates. Having found such a basis, we will then see that we can easily recover an  $H$ -conjugate basis. The Lanczos method is more usually thought of as a method for estimating eigenvalues, but this is a consequence of a rather different mechanism, Rayleigh–Ritz approximation. We must also consider the method from this perspective.

### 5.2.1 Computing an Orthonormal Basis for the Krylov Space

Our aim is to build an orthonormal basis  $\{q_0, q_1, \dots, q_k\}$  for the Krylov subspace  $\mathcal{K}(H, r_0, k)$ . We start with  $\mathcal{K}(H, r_0, 0) = \text{span}\{q_0\}$ , where  $q_0 = r_0/\|r_0\|_2$ . Thus

$$\mathcal{K}(H, r_0, j) = \mathcal{K}(H, q_0, j).$$

We suppose we have found a suitable basis

$$\mathcal{K}(H, q_0, i) = \text{span}\{q_0, q_1, \dots, q_i\}$$

for all  $0 \leq i \leq k$ , where the basis vectors  $q_i$  are mutually orthonormal, that is, where

$$\langle q_i, q_j \rangle = \delta_{i,j}$$

---

<sup>42</sup>It is almost always the case in numerical algorithms that if disaster occurs for a particular value of a parameter, numerical instability occurs for neighbouring values. A notable exception is, of course, inverse iteration for eigenproblems where close singularity is actively sought (Golub and Van Loan, 1989, p. 383).

and  $\delta_{i,j}$  is the Kronecker delta. We now aim to find a suitable  $q_{k+1}$ .

**Lemma 5.2.1** Suppose that

$$\mathcal{K}(H, q_0, i) = \text{span} \{q_0, q_1, \dots, q_i\} \quad (5.2.1)$$

for  $0 \leq i \leq k+1$ , and that the  $q_i$  are mutually orthonormal. Then

$$q_{i+1} \in \text{span} \{q_0, q_1, \dots, q_i, Hq_i\} \quad (5.2.2)$$

for all  $0 \leq i \leq k$ .

**Proof.** Firstly, by definition of the Krylov spaces and assumption (5.2.1),

$$\begin{aligned} \text{span} \{q_0, \dots, q_i, q_{i+1}\} &= \mathcal{K}(H, q_0, i+1) \\ &= \text{span} \{\mathcal{K}(H, q_0, i), H^{i+1}q_0\} \\ &= \text{span} \{q_0, \dots, q_i, H^{i+1}q_0\} \end{aligned}$$

for  $0 \leq i \leq k$ . Thus

$$q_{i+1} \in \text{span} \{q_0, q_1, \dots, q_i, H^{i+1}q_0\}, \quad (5.2.3)$$

and

$$H^{i+1}q_0 \in \text{span} \{q_0, q_1, \dots, q_{i+1}\} \quad (5.2.4)$$

for  $0 \leq i \leq k$ . The relationship (5.2.2) is trivially true when  $i = 0$ . Now assume, inductively, that

$$q_i \in \text{span} \{q_0, q_1, \dots, q_{i-1}, Hq_{i-1}\} \quad (5.2.5)$$

for  $0 \leq i \leq j$ . It then follows that

$$Hq_{i-1} \in \text{span} \{q_0, q_1, \dots, q_i\} \quad (5.2.6)$$

for  $0 \leq i \leq j$ , as the  $q_i$  are orthonormal. Then, following (5.2.3),

$$\begin{aligned} q_{j+1} &\in \text{span} \{q_0, q_1, \dots, q_j, H^{j+1}q_0\} \\ &= \text{span} \{q_0, q_1, \dots, q_j, H(H^j q_0)\} \\ &= \text{span} \{q_0, q_1, \dots, q_j, Hq_0, \dots, Hq_{j-1}, Hq_j\} \quad \text{from (5.2.4)} \\ &= \text{span} \{q_0, q_1, \dots, q_j, Hq_j\} \quad \text{from (5.2.6).} \end{aligned}$$

Thus, (5.2.5) holds for  $0 \leq j \leq k$ , which proves the lemma.  $\square$

The lemma says that the  $q_{k+1}$  we seek must lie in the span of the previous  $q_i$  and  $Hq_k$ . As  $q_{k+1}$  is required to be orthonormal to the previous  $q_i$ , it must contain a nonzero component of  $Hq_k$ . Thus it suffices to find a vector

$$y_{k+1} = \sum_{i=0}^k \theta_{ki} q_i + Hq_k \quad (5.2.7)$$

that is orthogonal to  $q_i$  for  $0 \leq i \leq k$ , and then to normalize  $y_{k+1}$  to give

$$q_{k+1} = y_{k+1}/\gamma_{k+1}, \quad (5.2.8)$$

where

$$\gamma_{k+1} = \|y_{k+1}\|_2.$$

At first sight, (5.2.7) does not look computationally attractive, as it appears to require that we have available all the previous  $q_i$ . Fortunately, appearances may be deceptive, as we shall now see. We must determine the coefficients  $\theta_{ki}$  in (5.2.7). We use the Gram–Schmidt process; that is, we form the inner product of (5.2.7) with  $q_j$  and use the orthogonality of  $y_{k+1}$  and  $q_j$  together with the orthonormality of the  $q_i$  to derive the relationship

$$0 = \langle q_j, y_{k+1} \rangle = \sum_{i=0}^k \theta_{ki} \langle q_j, q_i \rangle + \langle q_j, Hq_k \rangle = \theta_{kj} + \langle q_j, Hq_k \rangle. \quad (5.2.9)$$

As the  $q_i$  are orthonormal, (5.2.2) implies that

$$Hq_j \in \text{span} \{q_0, q_1, \dots, q_j, q_{j+1}\}$$

and hence that

$$\langle q_j, Hq_k \rangle = 0 \quad (5.2.10)$$

for all  $j \leq k - 2$ . Hence (5.2.9) and (5.2.10) imply that

$$\theta_{kj} = 0 \quad (5.2.11)$$

for all  $j \leq k - 2$ . Thus,  $y_{k+1}$  is defined solely as a linear combination of  $Hq_k$  and the previous two orthonormal basis vectors  $q_k$  and  $q_{k-1}$ . Combining (5.2.7), (5.2.8), (5.2.9), and (5.2.11) we therefore have that

$$y_{k+1} = \gamma_{k+1} q_{k+1} = Hq_k - \langle q_k, Hq_k \rangle q_k - \langle q_{k-1}, Hq_k \rangle q_{k-1}. \quad (5.2.12)$$

This relationship reveals an alternative definition of  $\gamma_{k+1}$ . For, taking the inner product of (5.2.12) with  $q_{k+1}$  and using the orthonormality of the  $q_i$  gives

$$\begin{aligned} \gamma_{k+1} &= \gamma_{k+1} \langle q_{k+1}, q_{k+1} \rangle \\ &= \langle q_{k+1}, Hq_k \rangle - \langle q_k, Hq_k \rangle \langle q_{k+1}, q_k \rangle - \langle q_{k-1}, Hq_k \rangle \langle q_{k+1}, q_{k-1} \rangle \\ &= \langle q_{k+1}, Hq_k \rangle. \end{aligned}$$

Thus  $\gamma_k = \langle q_k, Hq_{k-1} \rangle = \langle q_{k-1}, Hq_k \rangle$ , and we may write (5.2.12) as

$$y_{k+1} = \gamma_{k+1} q_{k+1} = Hq_k - \delta_k q_k - \gamma_k q_{k-1}, \quad (5.2.13)$$

where

$$\delta_k = \langle q_k, Hq_k \rangle.$$

In summary, we may compute orthonormal bases of the Krylov subspaces  $\mathcal{K}(H, r_0, k)$ ,  $k \geq 0$ , with the following algorithm.

**Algorithm 5.2.1: Lanczos method for an orthonormal basis of  $\mathcal{K}(H, r_0, k)$** 

Given  $r_0$ , set  $y_0 = r_0$ ,  $q_{-1} = 0$ , and, for  $k = 0, 1, \dots$ , perform the iteration

$$\begin{aligned}\gamma_k &= \|y_k\|_2, \\ q_k &= y_k/\gamma_k, \\ \delta_k &= \langle q_k, Hq_k \rangle, \\ y_{k+1} &= Hq_k - \delta_k q_k - \gamma_k q_{k-1}.\end{aligned}$$

In matrix terms, if we define  $Q_k = (q_0 \dots q_k)$  and take into account that  $q_{-1} = 0$ , then (5.2.13) is

$$HQ_k - Q_k T_k = \gamma_{k+1} q_{k+1} e_{k+1}^T, \quad (5.2.14)$$

where the tridiagonal matrix  $T_k$  is

$$T_k = \begin{pmatrix} \delta_0 & \gamma_1 & & & \\ \gamma_1 & \delta_1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{pmatrix} \quad (5.2.15)$$

and the columns of  $Q_k$  are orthonormal, that is,

$$Q_k^T Q_k = I \text{ and } Q_k^T q_{k+1} = 0. \quad (5.2.16)$$

Premultiplying (5.2.14) by  $Q_k^T$  and using (5.2.16) reveals the important relationship

$$Q_k^T HQ_k = T_k. \quad (5.2.17)$$

This equation forms the basis of the Rayleigh–Ritz procedure for computing eigenolutions of  $H$  that we shall discuss in Section 5.2.5.

### 5.2.2 Relationship with the Conjugate Direction Method

The Lanczos method computes an orthonormal basis  $\{q_0, q_1, \dots, q_k\}$  for the Krylov subspace  $\mathcal{K}(H, r_0, k)$ . The conjugate direction method indirectly does the same; for (5.1.27) shows that  $\mathcal{K}(H, r_0, k) = \text{span } \{r_0, r_1, \dots, r_k\}$ , while (5.1.21) shows that the  $r_i$  are orthogonal. Thus the vectors  $r_i/\|r_i\|_2$  form an orthonormal basis for  $\mathcal{K}(H, r_0, k)$ , and we must therefore have that

$$q_k = \sigma_k r_k / \|r_k\|_2, \quad (5.2.18)$$

where  $\sigma_k = \pm 1$ , for all  $k$  so long as the conjugate direction recurrence does not break down.

Given this seeming equivalence between the methods, it is natural to ask whether there is any stronger correspondence between the quantities computed by Algorithms 5.1.1 and 5.2.1. Substituting (5.1.24) and (5.1.26) into (5.1.24) and collecting terms, we have that

$$r_{k+1} = -\alpha_k H r_k + \left(1 + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}}\right) r_k - \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} r_{k-1}, \quad (5.2.19)$$

where we have artificially defined  $\beta_{-1} = 0$ . Similarly, substituting (5.2.18) into (5.2.13), using the relationship  $\sqrt{\beta_k} = \|r_{k+1}\|_2 / \|r_k\|_2$  from (5.1.25), normalizing, and defining  $\pi_k = \sigma_k / \sigma_{k+1} = \pm 1$ , we obtain

$$r_{k+1} = \frac{\pi_k \sqrt{\beta_k}}{\gamma_{k+1}} H r_k - \frac{\pi_k \delta_k \sqrt{\beta_k}}{\gamma_{k+1}} r_k - \frac{\pi_k \pi_{k-1} \gamma_k \sqrt{\beta_k \beta_{k-1}}}{\gamma_{k+1}} r_{k-1}. \quad (5.2.20)$$

Comparing coefficients of  $H r_k$ ,  $r_k$ , and  $r_{k-1}$  in (5.2.19) and (5.2.20), respectively, we see that

$$-\alpha_k = \pi_k \frac{\sqrt{\beta_k}}{\gamma_{k+1}}, \quad -\left(1 + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}}\right) = \frac{\pi_k \delta_k \sqrt{\beta_k}}{\gamma_{k+1}} \quad \text{and} \quad \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} = \frac{\pi_k \pi_{k-1} \gamma_k \sqrt{\beta_k \beta_{k-1}}}{\gamma_{k+1}}.$$

From this we deduce that  $\pi_k = -\text{sign}(\alpha_k)$  and hence  $\sigma_k = -\text{sign}(\alpha_{k-1})\sigma_{k-1}$  with  $\sigma_0 = 1$ , since both  $\gamma_{k+1}$  and  $\sqrt{\beta_k}$  are positive, and also that

$$\gamma_k = \frac{\sqrt{\beta_{k-1}}}{|\alpha_{k-1}|} \quad \text{and} \quad \delta_k = \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}}.$$

Thus, so long as the conjugate direction recurrence does not break down,

$$T_k = \begin{pmatrix} \frac{1}{\alpha_0} & \frac{\sqrt{\beta_0}}{|\alpha_0|} & & & & \\ \frac{\sqrt{\beta_0}}{|\alpha_0|} & \frac{1}{\alpha_1} + \frac{\beta_0}{\alpha_0} & \frac{\sqrt{\beta_1}}{|\alpha_1|} & & & \\ & \frac{\sqrt{\beta_1}}{|\alpha_1|} & \frac{1}{\alpha_2} + \frac{\beta_1}{\alpha_1} & \ddots & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \frac{1}{\alpha_{k-1}} + \frac{\beta_{k-2}}{\alpha_{k-2}} & \frac{\sqrt{\beta_{k-1}}}{|\alpha_{k-1}|} \\ & & & & \frac{\sqrt{\beta_{k-1}}}{|\alpha_{k-1}|} & \frac{1}{\alpha_k} + \frac{\beta_{k-1}}{\alpha_{k-1}} \end{pmatrix}, \quad (5.2.21)$$

and therefore the Lanczos tridiagonal matrix  $T_k$  is available as a trivial by-product of the conjugate direction (and conjugate gradient) method(s). We shall see in Section 5.2.5 how this enables us to estimate the condition number of  $H$ .

### 5.2.3 Finding Conjugate Directions from an Orthonormal Krylov Basis

Having found the desired orthonormal basis of  $\mathcal{K}(H, r_0, k)$ , we now consider how we may recover a suitable  $H$ -conjugate basis from it. When  $H$  is positive definite, the basis constructed by the conjugate direction method (Algorithm 5.1.1) is well defined, and the precaution of finding a conjugate basis via an orthonormal one is, in fact, unnecessary. But when  $H$  is, or might be, indefinite, and consequently when Algorithm 5.1.1 could break down, we really have little choice but to be cautious. We investigate both the positive definite and indefinite cases, the first setting the scene for the second.

### 5.2.3.1 Positive Definite $H$

If  $H$  is positive definite, (5.2.17) and the orthonormality of the columns of  $Q_k$  imply that  $T_k$  is also positive definite. Thus  $T_k$  may be stably decomposed as

$$T_k = L_k D_k L_k^T, \quad (5.2.22)$$

where

$$L_k = \begin{pmatrix} 1 & & & \\ \mu_0 & 1 & & \\ & \ddots & \ddots & \\ & & \mu_{k-1} & 1 \end{pmatrix} \text{ and } D_k = \begin{pmatrix} d_0 & & & \\ & d_1 & & \\ & & \ddots & \\ & & & d_k \end{pmatrix}. \quad (5.2.23)$$

If we expand the product (5.2.22) using (5.2.23), and compare it with (5.2.21), we immediately find that

$$d_i = \frac{1}{\alpha_i} \text{ and } \mu_i = \text{sign}(\alpha_i)\sqrt{\beta_i}$$

for  $0 \leq i \leq k$ . Equating (5.2.17) with (5.2.22), and pre- and postmultiplying, respectively, by the inverse of  $L_k$  and its transpose, we find that

$$S_k^T H S_k = D_k,$$

where we have defined

$$S_k = Q_k L_k^{-T}. \quad (5.2.24)$$

Thus, as  $D_k$  is diagonal, the columns of  $S_k$  form an  $H$ -conjugate basis for  $\mathcal{K}(H, r_0, k)$ . At first glance, it might appear that computing  $S_k$  from (5.2.24) is expensive. However, letting  $S_k = (s_1 \ s_2 \dots \ s_k)$ , writing (5.2.24) as

$$S_k L_k^T = Q_k,$$

and using the exact form of  $L_k$  in (5.2.23) reveals that

$$s_i = q_i - \mu_{i-1} s_{i-1} \quad (5.2.25)$$

for  $1 \leq i \leq k$ , where  $s_0 = q_0$ . Thus  $s_k$  may be computed very efficiently once we have obtained the new orthonormal basis vector  $q_k$ .

### 5.2.3.2 Indefinite $H$

When  $H$  is indefinite, the decomposition (5.2.22) may not exist. It is more likely, perhaps, for diagonal entries of  $D_k$  to be small, which would indicate potential numerical instability. We have seen this phenomenon before, in Section 4.3.4, and the cure here, as there, is to use a stable symmetric indefinite factorization.

We shall decompose  $T_k$  as

$$T_k = L_k B_k L_k^T, \quad (5.2.26)$$

where  $L_k$  is unit lower bidiagonal and  $B_k$  is block diagonal, with each block being of dimension at most 2.<sup>43</sup> Notice that the permutation matrix  $P$  involved in the general factorization (4.3.9) (p. 65) is missing from (5.2.26). This is possible because  $T_k$  is tridiagonal, and growth may be controlled simply by deciding at each stage whether the current potential diagonal pivot belongs in a 1 by 1 or 2 by 2 pivot block. The rule we use is that, if the current diagonal,  $b_d$ , has not already been assigned to a 2 by 2 pivot, it will be if and only if

$$|b_d| < \eta b_o^2, \quad (5.2.27)$$

where  $b_o$  is the subdiagonal directly below  $b_d$ , and  $\eta$  is a positive constant chosen to control growth.

Given the factorization (5.2.26), we now find eigendecompositions

$$B^{ii} = U^{ii} D^{ii} (U^{ii})^T$$

of each of the diagonal blocks that make up  $B_k$ . Here  $D^{ii}$  is a diagonal matrix of eigenvalues of  $B^{ii}$ , while the columns of  $U^{ii}$  are its normalized eigenvectors. Of course, as  $B^{ii}$  is of dimension at most 2, these eigensolutions are trivial to obtain. Having computed these, we construct the block diagonal matrix  $U_k$  by arranging the matrices  $U^{ii}$  along its diagonal, and similarly form  $D_k$  from the diagonal matrices  $D^{ii}$ . Thus we have

$$B_k = U_k D_k U_k^T$$

and hence

$$T_k = M_k D_k M_k^T, \quad (5.2.28)$$

where

$$M_k \stackrel{\text{def}}{=} L_k U_k = \begin{pmatrix} M^{00} & & & \\ M^{10} & M^{11} & & \\ & \ddots & \ddots & \\ & & \ddots & M^{j-1,j-1} \\ & & & M^{jj-1} & M^{jj} \end{pmatrix}. \quad (5.2.29)$$

The column and row dimensions of each subdiagonal block  $M^{i+1,i}$  are implied by those of the diagonal blocks  $M^{ii}$  and  $M^{i+1,i+1}$ , respectively. Moreover, the diagonal blocks  $M^{ii} = U^{ii}$  are orthonormal, and thus

$$(M^{ii})^{-1} = (U^{ii})^{-1} = (U^{ii})^T = (M^{ii})^T, \quad (5.2.30)$$

as the diagonal blocks in  $L_k$  are identity matrices and the  $U^{ii}$  are orthonormal.

Having found the decomposition (5.2.28), it is now straightforward to compute a set of  $H$ -conjugate directions. For, equating (5.2.17) with (5.2.28), and pre- and postmultiplying, respectively, by the inverse of  $M_k$  and its transpose, we find, as we did in Section 5.2.3.1, that

$$S_k^T H S_k = D_k,$$

---

<sup>43</sup>In addition, each of the corresponding diagonal blocks in  $L_k$  is an identity matrix.

where we now have defined

$$S_k = Q_k M_k^{-T}. \quad (5.2.31)$$

Thus, once again, as  $D_k$  is diagonal, the columns of  $S_k$  form an  $H$ -conjugate basis for  $\mathcal{K}(H, r_0, k)$ . To compute  $S_k$  efficiently, we let  $S_k = (S^0 \ S^1 \ \dots \ S^{j-1} \ S^j)$  and rewrite (5.2.31) as

$$S_k M_k^T = Q_k \stackrel{\text{def}}{=} (Q^0 \ Q^1 \ \dots \ Q^{j-1} \ Q^j). \quad (5.2.32)$$

Then, using the block bidiagonal form of  $M_k$  and the orthonormality of  $M^{ii}$ , it is straightforward to show that

$$S^i = (Q^i - S^{i-1}(M^{ii-1})^T) M^{ii} \quad (5.2.33)$$

for  $1 \leq i \leq j$ , where  $S^0 = Q^0 M^{00}$ . Thus, as before we may form a block of  $H$ -conjugate vectors  $S^j$  very efficiently once we have obtained  $Q^j$ ,  $M^{ii-1}$ , and  $M^{ii}$ . But now, stability considerations dictate that we may need to perform a pair of Lanczos iterations before we compute the new  $H$ -conjugate vectors.

#### 5.2.4 Approximation of Critical Points within the Subspace

One further question deserves an answer: Is it possible to use the Lanczos method directly<sup>44</sup> to compute the minimizer of a quadratic  $q$  within the Krylov space  $\mathcal{K}(H, g_0, k)$ ? The answer is “yes”, provided that  $H$  is positive definite. If a saddle point suffices, the answer is also “yes” for general  $H$ . Consider Lemma 5.1.1 in the case where  $S = Q_k$  and  $x = x_0$ . In this case, as

$$q_0 = \frac{g_0}{\|g_0\|_2} = \frac{g_0}{\gamma_0} \quad (5.2.34)$$

and  $Q_k^T q_0 = e_1$ , we have that

$$Q_k^T g_0 = \gamma_0 e_1. \quad (5.2.35)$$

Thus (5.1.2) gives that

$$x_{k+1} = x_0 + Q_k h_k, \quad \text{where } T_k h_k = -\gamma_0 e_1. \quad (5.2.36)$$

Suppose that  $T_k$  is given by (5.2.28) and that  $S_k$  satisfies (5.2.31). This covers both the case of indefinite  $H$  considered in Section 5.2.3.2 and, by insisting that all blocks are of dimension 1, the special case where  $H$  is positive definite considered in Section 5.2.3.1. Then (5.2.36) may be expressed as

$$x_{k+1} = x_0 - S_k D_k^{-1} M_k^{-1}(\gamma_0 e_1)$$

using (5.2.32). Suppose further that the  $k$ th iteration completes the  $j$ th block of the decomposition (5.2.28), while the  $l$ th iteration completed block  $j-1$ , and that we write  $x_{k+1} \equiv x^{j+1}$  and  $x_{l+1} \equiv x^j$ . Then

$$x^{j+1} = x_0 - S_k D_k^{-1} M_k^{-1}(\gamma_0 e_1) \quad \text{and} \quad x^j = x_0 - S_l D_l^{-1} M_l^{-1}(\gamma_0 e_1),$$

---

<sup>44</sup>It is, of course, possible to compute the minimizer indirectly by using the set of conjugate directions computed from the orthonormal Krylov basis in Section 5.2.3.

and hence

$$x^{j+1} = x^j - S_k D_k^{-1} M_k^{-1}(\gamma_0 e_1) + S_l D_l^{-1} M_l^{-1}(\gamma_0 e_1). \quad (5.2.37)$$

But now, finding  $z \stackrel{\text{def}}{=} -M_k^{-1}(\gamma_0 e_1)$  is equivalent to solving

$$\begin{pmatrix} M^{00} & & & \\ M^{10} & M^{11} & & \\ \cdot & \cdot & \ddots & \\ & & \ddots & M^{j-1,j-1} \\ & & & M^{jj-1} & M^{jj} \end{pmatrix} \begin{pmatrix} z^0 \\ z^1 \\ \vdots \\ z^{j-1} \\ z^j \end{pmatrix} = - \begin{pmatrix} \gamma_0 e_1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \quad (5.2.38)$$

where the dimensions of the components  $z^i$  are those of the blocks  $D^i$ . Thus, letting  $S^i$  be the block columns of  $S_k$  and  $D^i$  be the block diagonals of  $D_k$ , we may rewrite (5.2.37) as

$$x^{j+1} = x^j + \sum_{i=0}^j S^i(D^i)^{-1} z^i - \sum_{i=0}^{j-1} S^i(D^i)^{-1} z^i = x^j + S^j(D^j)^{-1} z^j,$$

while the last row block of (5.2.38) and the orthonormality of each  $M^{ii}$  reveal that

$$z^j = -(M^{jj})^T M^{jj-1} z^{j-1},$$

where  $z^0 = -\gamma_0(M^{00})^T e_1$ . When  $H$  is positive definite, this recurrence is simply

$$x_{k+1} = x_k + (z_k/d_k)s_k, \quad \text{where } z_k = -z_{k-1}\mu_{k-1}$$

and  $z_0 = -\gamma_0$ . Notice that these recurrences require  $s_k$  or  $S_k$ , which are themselves available by recurrence from (5.2.25) and (5.2.33).

It is also easy to monitor the gradient of  $q$  at  $x_{k+1}$ , for

$$\begin{aligned} g(x_{k+1}) &= Hx_{k+1} + c \\ &= H(x_{k+1} - x_0) + g_0 \\ &= HQ_k h_k + \gamma_0 q_0 && \text{from (5.2.34) and (5.2.36)} \\ &= HQ_k h_k + \gamma_0 Q_k e_1 \\ &= HQ_k h_k - Q_k T_k h_k && \text{from (5.2.36)} \\ &= \gamma_{k+1} \langle e_{k+1}, h_k \rangle q_{k+1} && \text{from (5.2.14).} \end{aligned} \quad (5.2.39)$$

It immediately follows that

$$\|g(x_{k+1})\|_2 = \gamma_{k+1} |\langle e_{k+1}, h_k \rangle| \quad (5.2.40)$$

as  $\|q_{k+1}\|_2 = 1$ , and thus  $x_{k+1}$  is a critical point for  $q$  whenever  $\gamma_{k+1} = 0$ . To obtain either (5.2.39) or (5.2.40) in a convenient form, we require  $\langle e_{k+1}, h_k \rangle$ . Fortunately, this is easy to obtain, as (5.2.36) and (5.2.28) give

$$\langle e_{k+1}, h_k \rangle = -\langle e_{k+1}, T_k^{-1}(\gamma_0 e_1) \rangle = -\langle M_k^{-1} e_{k+1}, D_k^{-1} M_k^{-1}(\gamma_0 e_1) \rangle,$$

while (5.2.29), (5.2.30), and (5.2.38) imply that

$$M_k^{-1}e_{k+1} = \begin{pmatrix} 0 \\ 0 \\ \cdot \\ 0 \\ (U^{jj})^T e_{\dim \mathbf{B}^{jj}} \end{pmatrix} \quad \text{and} \quad M_k^{-1}(\gamma_0 e_1) = - \begin{pmatrix} z^0 \\ z^1 \\ \cdot \\ z^{j-1} \\ z^j \end{pmatrix},$$

respectively. Thus

$$\langle e_{k+1}, h_k \rangle = \langle (D^{jj})^{-1} z^j, (U^{jj})^T e_{\dim \mathbf{B}^{jj}} \rangle = \langle z^j, (D^{jj})^{-1} (U^{jj})^T e_{\dim \mathbf{B}^{jj}} \rangle.$$

When  $H$  is positive definite, this simplifies to give

$$\langle e_{k+1}, h_k \rangle = z_k/d_k.$$

### 5.2.5 Rayleigh–Ritz Approximations to Eigenpairs

Although, as we have seen, the Lanczos method may be used to solve symmetric linear systems,<sup>45</sup> it is better known as a method for finding a few eigenvalues of a large symmetric matrix. In this section, we briefly examine the method from this perspective.

Given a symmetric  $n$  by  $n$  matrix  $H$  and an  $n$  by  $k$  ( $k \leq n$ ) matrix  $Q_k$  with orthonormal columns, the eigenvalues and eigenvectors of the  $k$  by  $k$  matrix

$$R_k = Q_k^T H Q_k$$

are known as *Ritz values* and *Ritz vectors*, respectively. If  $Q_{k+1}$ ,  $k < n$ , is formed by appending a further orthonormal column  $q_{k+1}$  to  $Q_k$ , the Ritz values of  $R_k$  interlace those of  $R_{k+1}$ , that is,

$$\lambda_1[R_{k+1}] \leq \lambda_1[R_k] \leq \lambda_2[R_{k+1}] \leq \cdots \leq \lambda_k[R_{k+1}] \leq \lambda_k[R_k] \leq \lambda_{k+1}[R_{k+1}].$$

As  $R_n$  is similar to  $H$ , we therefore see that the Ritz values of  $R_k$  converge outwards to the eigenvalues of  $H$  as  $k$  increases to  $n$ . The Ritz pair  $(u_j[R_k], \lambda_j[R_k])$  satisfies the bound

$$|\lambda_i[H] - \lambda_j[R_k]| \leq \|Hv_j[R_k] - \lambda_j[R_k]v_j[R_k]\|_2, \quad (5.2.41)$$

where<sup>46</sup>

$$v_j[R_k] = Q_k u_j[R_k] \quad (5.2.42)$$

for some eigenvalue  $\lambda_i[H]$  of  $H$ , and thus if  $\|Hv_j[R_k] - \lambda_j[R_k]v_j[R_k]\|_2$  is small,  $\lambda_j[R_k]$  must be close to an eigenvalue of  $H$ . The Rayleigh–Ritz procedure is simply any method that obtains a  $Q_k$  with orthonormal columns, finds the Ritz pairs associated

<sup>45</sup>In our case, the system in question is  $Hx + c = 0$ .

<sup>46</sup>This is a special case of a very general, but remarkably simple, result that, for any value  $\sigma$  and unit vector  $v$ , there is an eigenvalue  $\lambda$  of  $H$  for which  $|\lambda - \sigma| \leq \|Hv - \sigma v\|$ . See, for example, Parlett (1980, Theorem 4.5.1).

with  $R_k$ , and uses estimates like (5.2.41) to determine how close the Ritz pairs are to eigenpairs of  $H$ . The procedure has been shown to be an optimal process, given the limited information available from  $Q_k$  and  $R_k$ , from a number of points of view.

Clearly, the dominant cost here is likely to be the computation of  $R_k$  and its associated Ritz pairs. If, however, we obtain  $Q_k$  from the Lanczos process, (5.2.17) shows that  $R_k$  is the tridiagonal matrix  $T_k$ . There are extremely effective methods for finding selected eigenpairs of such a matrix, and further economies may be made because  $T_k$  is the leading submatrix of  $T_{k+1}$ .

If  $\gamma_{k+1}$  is ever zero, (5.2.14) shows that  $HQ_k = Q_kT_k$ , and thus we have discovered an *invariant subspace* of  $H$ ; the resulting Ritz values are eigenvalues of  $H$ , while the associated eigenvectors (5.2.42) may be found from the Ritz vectors so long as  $Q_k$  can be recovered or has been stored.<sup>47</sup>

The test (5.2.41) appears to require that we can compute  $v_j[T_k]$ , which, as we have just implied, may be expensive. Fortunately, for the Lanczos method, there is a cheap alternative, for

$$\begin{aligned} \|Hv_j[T_k] - \lambda_j[T_k]v_j[T_k]\|_2 &= \|HQ_ku_j[T_k] - \lambda_j[T_k]Q_ku_j[T_k]\|_2 && \text{from (5.2.42)} \\ &= \|(HQ_k - Q_kT_k)u_j[T_k]\|_2 && \text{as } (u_j[T_k], \lambda_j[T_k]) \text{ is a Ritz pair of } T_k \\ &= \|\gamma_{k+1}\langle e_{k+1}, u_j[T_k] \rangle q_{k+1}\|_2 && \text{from (5.2.14)} \\ &= \gamma_{k+1}|\langle e_{k+1}, u_j[T_k] \rangle| && \text{as } \|q_{k+1}\|_2 = 1. \end{aligned}$$

Hence we may replace (5.2.41) by

$$|\lambda_i[H] - \lambda_j[T_k]| \leq \gamma_{k+1}|\langle e_{k+1}, u_j[T_k] \rangle|.$$

Note that  $|\langle e_{k+1}, u_j[T_k] \rangle| \leq 1$  as the Ritz vectors have been normalized, and thus a small  $\gamma_{k+1}$  necessarily implies that the Ritz values are close to eigenvalues of  $H$ . It is important to note that a Ritz value may also be close if the last component of its associated Ritz vector is small, even when  $\gamma_{k+1}$  is large.

One further use of the Lanczos method is to estimate the condition number of  $H$ . Clearly the ratio of the largest to smallest Ritz values provides a lower bound when  $H$  is positive definite, and there are good theoretical reasons to believe that it is these extreme Ritz values that converge first. An explanation is unfortunately beyond the scope of this book.

### 5.2.6 Preconditioned Lanczos

We have seen that the conjugate gradient and Lanczos methods produce a number of useful but different bases for the underlying Krylov space. However, by preference, we would normally anticipate using the *preconditioned* conjugate gradient method rather than its unpreconditioned predecessor, and this of course generates a different Krylov space. It goes without saying that there is also a preconditioned variant of the Lanczos

---

<sup>47</sup>A second pass may be needed here to recompute  $Q_k$ , or rather the columns of  $Q_k$ , one by one. Alternatively,  $Q_k$  might have been written to backup storage.

method, and this produces a better behaved basis for this new space, that is, better behaved than preconditioned conjugate gradients. In this section, we shall briefly consider this method and its properties.

As we have seen, the preconditioned conjugate direction method investigates the Krylov space

$$\mathcal{K}(M^{-1}H, M^{-1}r_0, k) = R^{-1}\mathcal{K}(\bar{H}, \bar{r}_0, k), \quad (5.2.43)$$

where, as in Section 5.1.6,

$$M = R^T R, \quad \bar{H} = R^{-T} H R^{-1}, \quad \text{and} \quad \bar{r}_0 = R^{-T} r_0.$$

The preconditioned Lanczos method then aims to build an orthonormal basis,  $\{q_0, q_1, \dots, q_k\}$ , for  $\mathcal{K}(\bar{H}, \bar{r}_0, k)$ , using the following appropriately transformed version of Algorithm 5.2.1.

**Algorithm 5.2.2: Lanczos method for an orthonormal basis of  $\mathcal{K}(\bar{H}, \bar{r}_0, k)$**

Given  $\bar{r}_0$ , set  $\bar{y}_0 = \bar{r}_0$ ,  $\bar{q}_{-1} = 0$  and, for  $k = 0, 1, \dots$ , perform the iteration

$$\begin{aligned} \gamma_k &= \|\bar{y}_k\|_2, \\ \bar{q}_k &= \bar{y}_k / \gamma_k, \\ \delta_k &= \langle \bar{q}_k, \bar{H} \bar{q}_k \rangle, \\ \bar{y}_{k+1} &= \bar{H} \bar{q}_k - \delta_k \bar{q}_k - \gamma_k \bar{q}_{k-1}. \end{aligned} \quad (5.2.44)$$

The generated  $\bar{q}_k$  are then orthogonal. To remove the dependence on  $\bar{H}$ , we simply define

$$q_i = R^{-1} \bar{q}_i \quad \text{and} \quad y_i = R^{-1} \bar{y}_i \quad (5.2.45)$$

and premultiply (5.2.44) by  $R^{-1}$ . This then immediately gives the following method.

**Algorithm 5.2.3: Preconditioned Lanczos method—preliminary version**

Given  $r_0$ , set  $y_0 = M^{-1}r_0$ ,  $q_{-1} = 0$  and, for  $k = 0, 1, \dots$ , perform the iteration

$$\begin{aligned} \gamma_k &= \sqrt{\langle y_k, M y_k \rangle}, \\ q_k &= y_k / \gamma_k, \\ \delta_k &= \langle q_k, H q_k \rangle, \\ y_{k+1} &= M^{-1} H q_k - \delta_k q_k - \gamma_k q_{k-1}. \end{aligned}$$

Note that Algorithm 5.2.3 rather inconveniently appears to require both  $M$  and its inverse. This need not be the case. For suppose we define

$$w_i = Mq_i \text{ and } t_i = My_i.$$

This, then, defines the preconditioned Lanczos method.

**Algorithm 5.2.4: Preconditioned Lanczos method**

Given  $r_0$ , set  $t_0 = r_0$ ,  $w_{-1} = 0$  and, for  $k = 0, 1, \dots$ , perform the iteration

$$y_k = M^{-1}t_k, \quad (5.2.46)$$

$$\gamma_k = \sqrt{\langle t_k, y_k \rangle}, \quad (5.2.47)$$

$$w_k = t_k / \gamma_k,$$

$$q_k = y_k / \gamma_k, \quad (5.2.48)$$

$$\delta_k = \langle q_k, Hq_k \rangle, \quad (5.2.49)$$

$$t_{k+1} = Hq_k - \delta_k w_k - \gamma_k w_{k-1}. \quad (5.2.50)$$

Notice that it is the  $\bar{q}_k$ , not the  $q_k$ , which are orthonormal. Indeed, if we let

$$Q_k = (q_0 \cdots q_k),$$

then we immediately deduce from (5.2.16) (applied to  $\bar{Q}_k$ ) and (5.2.45) that

$$Q_k^T M Q_k = I \text{ and } Q_k^T M q_{k+1} = 0. \quad (5.2.51)$$

The fundamental Lanczos relationship (5.2.14) becomes

$$H Q_k - M Q_k T_k = \gamma_{k+1} w_{k+1} e_{k+1}^T, \quad (5.2.52)$$

where the tridiagonal matrix  $T_k$  is of the form (5.2.15), but with  $\gamma_i$  and  $\delta_i$  now defined by (5.2.47) and (5.2.49). The analog of (5.2.17) is

$$Q_k^T H Q_k = T_k. \quad (5.2.53)$$

Finally, as  $\text{span}\{\bar{q}_0, \bar{q}_1, \dots, \bar{q}_k\}$  forms an orthogonal basis for  $\mathcal{K}(\bar{H}, \bar{r}_0, k)$ , (5.2.43) and (5.2.45) show that

$$\mathcal{K}(M^{-1}H, M^{-1}r_0, k) = R^{-1} \text{span}\{\bar{q}_0, \bar{q}_1, \dots, \bar{q}_k\} = \text{span}\{q_0, q_1, \dots, q_k\}.$$

Therefore,  $\{q_0, q_1, \dots, q_k\}$  forms an  $M$ -orthonormal basis for  $\mathcal{K}(M^{-1}H, M^{-1}r_0, k)$ , the Krylov space generated by the preconditioned conjugate direction method.

As one would expect from Section 5.2.2, there is a strong relationship between the preconditioned conjugate gradient and Lanczos methods. In particular, the preconditioned Lanczos tridiagonal  $T_k$  satisfies (5.2.21) for the  $\alpha_i$  and  $\beta_i$  generated by the

preconditioned conjugate gradient method. In addition, the vectors  $g_i$  generated by the preconditioned conjugate gradient method are  $M$ -orthogonal, and therefore

$$q_j = \pm M^{-1}g_j/\sqrt{\langle g_j, M^{-1}g_j \rangle} \text{ and } w_j = \pm g_j/\sqrt{\langle g_j, M^{-1}g_j \rangle}$$

for  $0 \leq j \leq k$ , so long as the iteration does not break down. If one needs to compute  $\bar{H}$ -conjugate vectors from the preconditioned Lanczos basis  $\{q_0, q_1, \dots, q_k\}$ , the methods considered in Section 5.2.3 may be used without change, so long as one uses the  $T_k$  and  $Q_k$  computed from Algorithm 5.2.4. Likewise, the recurrences given in Section 5.2.4 for finding a critical point of  $q$  are appropriate. The only minor change is that the formulae (5.2.39) and (5.2.40) for the gradient and its norm should be replaced by

$$g(x_{k+1}) = \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} \text{ and } \|g(x_{k+1})\|_{M^{-1}} = \gamma_{k+1}|\langle e_{k+1}, h_k \rangle|. \quad (5.2.54)$$

It is important for later to note that the identity

$$Q_k^T g_0 = \gamma_0 e_1 \quad (5.2.55)$$

continues to hold.

Turning to the other use of the Lanczos method, if  $(u_j[T_k], \lambda_j[T_k])$  is an eigenpair of  $T_k$ , (5.2.52) implies that

$$\begin{aligned} HQ_k u_j[T_k] - MQ_k T_k u_j[T_k] &= HQ_k u_j[T_k] - \lambda_j[T_k] MQ_k u_j[T_k] \\ &= \gamma_{k+1}\langle e_{k+1}, u_j[T_k] \rangle M q_{k+1}, \end{aligned}$$

and thus that  $(Q_k u_j[T_k], \lambda_j[T_k])$  is an approximate generalized eigenpair of  $(H, M)$  provided that  $\gamma_{k+1}$  or  $\langle e_{k+1}, u_j[T_k] \rangle$  are small. We shall see later in the book that it is sometimes useful to obtain such a generalized eigenpair, and thus the Lanczos variant of the preconditioned conjugate gradient method plays a second important role.

## Notes and References for Section 5.2

The Lanczos (1950) method was originally proposed to determine eigenvalues of linear operators. Good descriptions are given by Parlett (1980, Chapter 13) and Stoer (1983, Section 2). The equivalence between the conjugate gradient and Lanczos methods is investigated by Cullum and Willoughby (1980) and Golub and Van Loan (1989, Section 9.3.1). There have been a number of methods that aim to construct conjugate directions from the Lanczos basis. The method we describe in Section 5.2.3 is essentially the SYMMBK (symmetric Bunch–Kaufman) method of Chandra (1978, Section 9.4). The pivot strategy (5.2.27), along with suitable choices of  $\eta$  and a justification that it suffices to control growth, is described by Bunch (1974). Other variants, such as the SYMMLQ (symmetric LQ) method of Paige and Saunders (1975), and the STOD (stabilized orthogonal direction) method of Stoer (1983), have been proposed.

We have painted a more optimistic picture of the method than it may deserve, since the process is extremely sensitive to computer roundoff. In particular, there may be a serious loss of orthogonality in the computed  $q_i$ . Fortunately, this difficulty does not imply that the

computed Ritz values are inaccurate, merely that the floating-point algorithm does not behave like its exact counterpart. The loss of orthogonality in the  $q_i$  typically only occurs as a Ritz value converges to an eigenvalue. In theory, it is highly likely that the extreme eigenvalues are discovered first. Thus, if it is only these values that are required, the subsequent floating-point behaviour is of little concern. When interior eigenvalues are required, it may be necessary to reorthogonalize the computed  $q_i$  at various stages of the computation. Explanations and remedies are given by Parlett (1980, Chapter 13) and are based on pioneering work by Kaniel (1966) and Paige (1971). An analysis of the method as a means to calculate critical points of a quadratic form is considered by Druskin, Greenbaum, and Knizhnerman (1998), who show that the method is still effective in this respect despite the loss of orthogonality of the Lanczos vectors.

The preliminary version of the preconditioned Lanczos method may be found in Nash (1984, Section 2.6), while the preferred version is given by Parlett (1980, Section 15.11).

### 5.3 Linear Least-Squares Problems

One problem that occurs frequently in practice is the linear least-squares problem

$$\underset{x \in \mathbb{R}^m}{\text{minimize}} \quad \|A^T y - g\|_2, \quad (5.3.1)$$

which we considered in Section 4.4.1. When the number of unknowns is large, an obvious approach is to apply the (preconditioned) conjugate gradient method to the equivalent problem

$$\underset{y \in \mathbb{R}^m}{\text{minimize}} \quad \frac{1}{2} \langle y, AA^T y \rangle - \langle y, Ag \rangle. \quad (5.3.2)$$

In this case, we have the following interesting result.

**Theorem 5.3.1** Suppose we apply Algorithm 5.1.2 to the problem (5.3.2), and that the initial point lies in the range-space of  $A$ ,  $\mathcal{R}(A)$ . Then all the generated iterates remain in this space.

**Proof.** We have that  $g_0 = AA^T y_0 - Ag = A(A^T y_0 - c)$  and  $p_0 = -g_0$ , and hence that  $y_0$ ,  $g_0$ , and  $p_0$  all lie in  $\mathcal{R}(A)$ . Suppose now that  $y_k$ ,  $g_k$ , and  $p_k$  all lie in this space, that is, that

$$y_k = Ay_k^R, \quad g_k = Ag_k^R, \quad \text{and} \quad p_k = Ap_k^R$$

for some appropriate vectors  $y_k^R$ ,  $g_k^R$ , and  $p_k^R$ . Then it follows from the algorithm that

$$y_{k+1} = y_k + \alpha_k p_k = Ay_k^R + \alpha_k Ap_k^R = A(y_k^R + \alpha_k p_k^R),$$

which shows that  $y_{k+1}$  lies in  $\mathcal{R}(A)$ . Furthermore, the algorithm also sets

$$g_{k+1} = g_k + \alpha_k AA^T p_k = Ag_k^R + \alpha_k AA^T Ap_k^R = A(g_k^R + \alpha_k A^T Ap_k^R)$$

and thus  $g_{k+1} \in \mathcal{R}(A)$ . Finally, the algorithm assigns

$$p_{k+1} = -g_{k+1} + \beta_k p_k = -Ag_{k+1}^R + \beta_k Ap_k^R = A(-g_{k+1}^R + \beta_k p_k^R),$$

from which we see that  $p_{k+1} \in \mathcal{R}(A)$ . Thus the theorem is true by induction.  $\square$

Of particular interest is the case when the initial value  $y_0 = 0$ . In this case, the conjugate gradient method converges to the least-squares solution of *minimum  $\ell_2$  norm*. This will be particularly important in later sections, where the conjugate gradient method is applied to the *underdetermined least-squares problem*.

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \|As + c\|_2, \quad (5.3.3)$$

where  $m \leq n$ . If the equations  $As + c = 0$  are consistent, all minimizers of (5.3.3) give the value zero, but the conjugate gradient method will give the minimizer of minimum norm. Finally, we note that when a preconditioner  $M$  is used, the conjugate gradient method converges to the minimum  $M$ -norm solution.

In practice, the recurrences in Algorithm 5.1.2 are usually rearranged to give the following special least-squares conjugate gradient method for (5.3.1).

**Algorithm 5.3.1: The conjugate gradient least-squares method**

Given  $y_0$ , set  $r_0 = A^T y_0 - g$ , and let  $g_0 = Ar_0$  and  $p_0 = -g_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\begin{aligned} q_k &= A^T p_k, \\ \alpha_k &= \|g_k\|_2^2 / \|q_k\|_2^2, \\ x_{k+1} &= x_k + \alpha_k p_k, \\ r_{k+1} &= r_k + \alpha_k q_k, \\ g_{k+1} &= Ar_{k+1}, \\ \beta_k &= \|g_{k+1}\|_2^2 / \|g_k\|_2^2, \\ p_{k+1} &= -g_{k+1} + \beta_k p_k. \end{aligned}$$

The residual vector  $r_k$  here is simply the vector  $g_k^R$  we saw in the proof of Theorem 5.3.1.

### Notes and References for Subsection 5.3.0

See Björck (1996, Chapter 7) for full details of conjugate gradient methods for linear least squares. There are also Lanczos-like variants that are slightly more reliable when the problem is ill-conditioned. See Paige and Saunders (1982).

## 5.4 Problems with Constraints

The final topic of interest is a generalization of that we considered in Section 5.1. Here we wish to minimize a quadratic function, but now we wish to restrict the solution to satisfy a set of linear constraints. Since, as we saw in Section 4.4.2, it is normally relatively easy to find a point that satisfies the constraints, we may assume that we have such a feasible point, and thus may merely require that the solution satisfy the affine constraints  $Ax = 0$ . To simplify matters further, we shall assume that  $A$  is  $m$  by  $n$  ( $m \leq n$ ) and that  $A$  is of full rank. Thus we are interested in the linearly constrained, quadratic minimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad q(x) = \frac{1}{2} \langle x, Hx \rangle + \langle c, x \rangle \quad \text{subject to } Ax = 0. \quad (5.4.1)$$

Since the constraints are affine, we have that  $x = Nx^N$ , where the columns of  $N$  form a basis for the null-space of  $A$  (see Section 4.4.2 for details of how we might find such a basis). Therefore, (5.4.1) is equivalent to solving the *unconstrained* minimization problem

$$\underset{x^N \in \mathbb{R}^{n-m}}{\text{minimize}} \quad q(Nx^N) = \frac{1}{2} \langle x^N, H^N x^N \rangle + \langle c^N, x^N \rangle, \quad (5.4.2)$$

where  $H^N = N^T H N$  and  $c^N = N^T c$ . We see immediately that (5.4.1) does not have a finite solution if  $H^N$  is indefinite, and shall restrict our attention to the case in which  $H^N$  is positive definite.

The iterative methods given in Section 5.1 are then entirely appropriate, save for two issues, namely, the computation and application of  $N$ , and the preconditioning of  $H^N$ . While it may be rather expensive to compute an orthonormal basis  $N$ , a nonorthonormal one of the form (4.4.13) (p. 72) is often inexpensive, particularly as iterative methods merely need to form products involving  $H^N$ , and thus  $N$  and  $N^T$ . Such products do not require that we form  $N$  or its transpose, merely that we can recover the action of these matrices on a given vector. But then it is quite clear that this merely requires that we can form products with  $A^N$  and its transpose and solve linear systems with the coefficient matrix  $A^R$  and its transpose. Thus, so long as a sparse factorization of  $A^R$  is practical, the conjugate gradient method may be applied.

The computation of a good preconditioner is an entirely different matter for two reasons. Firstly, it is unreasonable in general to expect that we can form  $H^N$ , both because of the work involved and because it is very likely to be a relatively dense matrix. Thus, we are unlikely to be able to draw much useful spectral or structural information from  $H^N$ , which is often crucial for a successful preconditioner. Secondly, since  $H^N$  depends on  $N$ , a bad choice for  $N$  may result in a very badly conditioned problem, often far worse than the natural conditioning of (5.4.1). Thus, preconditioning becomes all the more important. Hence, it is of interest to discover if there are alternative iterative methods, which both avoid the need for  $N$ , and for which preconditioning may be more transparent. This is the topic of this section.

### 5.4.1 Projected Preconditioned Conjugate Gradients

In order to derive a suitable method for (5.4.1) we first recall what Algorithm 5.1.4 would be if applied to (5.4.2). We suppose that  $M^N$  is some easily invertible approximation to  $H^N$ . This leads to Algorithm 5.4.1.

#### Algorithm 5.4.1: Preconditioned conjugate gradients for (5.4.2)

Given  $x_0$ , set  $g_0^N = N^T H N x_0 + N^T c$ , and let  $v_0^N = (M^N)^{-1} g_0^N$  and  $p_0^N = -v_0^N$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\begin{aligned}\alpha_k &= \langle g_k^N, v_k^N \rangle / \langle p_k^N, N^T H N p_k^N \rangle, \\ x_{k+1}^N &= x_k^N + \alpha_k p_k^N, \\ g_{k+1}^N &= g_k^N + \alpha_k N^T H N p_k^N, \\ v_{k+1}^N &= (M^N)^{-1} g_{k+1}^N, \\ \beta_k &= \langle g_{k+1}^N, v_{k+1}^N \rangle / \langle g_k^N, v_k^N \rangle, \\ p_{k+1}^N &= -v_{k+1}^N + \beta_k p_k^N.\end{aligned}$$

Recover  $x = Nx_{k+1}^N$ .

While the above algorithm is expressed in terms of variables that lie in the null-space of  $A$ , it is easy to translate the algorithm back into the original variables. To do this, we define  $n$ -vectors  $x, g, v, p$ , which satisfy  $x = Nx^N$ ,  $N^T g = g^N$ ,  $v = Nv^N$ , and  $p = Np^N$ . Substituting these into Algorithm 5.4.1 leads to Algorithm 5.4.2 for solving (5.4.1).

#### Algorithm 5.4.2: Preconditioned conjugate gradients for (5.4.1)

Given  $x_0$  for which  $Ax_0 = 0$ , set  $g_0 = Hx_0 + c$ , and let

$$v_0 = N(M^N)^{-1} N^T g_0 \quad (5.4.3)$$

and  $p_0 = -v_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration

$$\begin{aligned}\alpha_k &= \langle g_k, v_k \rangle / \langle p_k, H p_k \rangle, \\ x_{k+1} &= x_k + \alpha_k p_k, \\ g_{k+1} &= g_k + \alpha_k H p_k, \\ v_{k+1} &= N(M^N)^{-1} N^T g_{k+1}, \\ \beta_k &= \langle g_{k+1}, v_{k+1} \rangle / \langle g_k, v_k \rangle, \\ p_{k+1} &= -v_{k+1} + \beta_k p_k.\end{aligned} \quad (5.4.4)$$

Finding an initial value  $x_0$  for which  $Ax_0 = 0$  is, of course, extremely simple since  $x_0 = 0$  is suitable. The only remaining complication is the computation of  $v_{k+1}$  in (5.4.4). A significant simplification occurs if we recognize that, since  $M^N$  is supposed to approximate  $H^N$ , we might equally require that  $M^N$  has the form

$$M^N = N^T MN,$$

where  $M$  should approximate  $H$ . In this case, (5.4.4) becomes

$$v_{k+1} = N(N^T MN)^{-1} N^T g_{k+1}.$$

While this does not, at first glance, appear to be a significant advance, we see that it is once we recognize that this is simply the null-space method for solving the equality-constrained quadratic program

$$\underset{v \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle v, Mv \rangle - \langle v, g_{k+1} \rangle \quad \text{subject to } Av = 0$$

(see Section 4.4.2), and thus that  $v_{k+1}$  also satisfies the augmented system

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} v_{k+1} \\ w_{k+1} \end{pmatrix} = \begin{pmatrix} g_{k+1} \\ 0 \end{pmatrix} \quad (5.4.5)$$

for some auxiliary vector  $w_{k+1}$ . Thus, in practice, we use Algorithm 5.4.2, but replace (5.4.3)/(5.4.4) by (5.4.5). The resulting method is known as the *projected preconditioned conjugate gradient* method. Of course, we solve (5.4.5) using one of the stable symmetric indefinite factorization methods discussed in Section 4.3.4.

Notice that Theorem 3.2.4 (p. 40) implies that the solution to (5.4.1) satisfies the augmented system

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ -y \end{pmatrix} = \begin{pmatrix} -c \\ 0 \end{pmatrix}. \quad (5.4.6)$$

Therefore Algorithm 5.4.2 is only appropriate if the cost of solving (5.4.6) is significantly greater than that for a sequence of (5.4.5), and the art is in choosing the preconditioner so that this is so. If  $M$  is diagonal, the range- or null-space approaches of Section 4.4.2 are sometimes preferable to (5.4.5), illustrating the flexibility of Algorithm 5.4.2.

We must enter a word of caution here. While Algorithm 5.4.2 may appear to be attractive, it is crucial that (5.4.5) be solved accurately, as otherwise, recurrences which rely on  $v_{k+1}$  lying in the null-space of  $A$  may be invalid. A particularly troublesome case occurs when  $g_{k+1}$  is large but  $v_{k+1}$  is small, for then (5.4.5) indicates that  $w_{k+1}$  will usually be large—such cases often occur in SQP methods (see Chapter 15) when approaching the solution of a nonlinear program. In this case, although it is possible to compute the composite vector  $(v_{k+1} \ w_{k+1})$  to high relative accuracy provided a stable factorization is used, the components  $v_{k+1}$  may have little relative accuracy. A number of precautions, including iterative refinement (see the last paragraph on p. 111), have been proposed, but the most effective appears to be to note that (5.4.5) is equivalent to

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} v_{k+1} \\ u_{k+1} \end{pmatrix} = \begin{pmatrix} g_{k+1} - A^T y_{k+1} \\ 0 \end{pmatrix}, \quad (5.4.7)$$

where  $w_{k+1} = y_{k+1} + u_{k+1}$ , and to choose  $y_{k+1}$  so that  $\|g_{k+1} - A^T y_{k+1}\|$  is small. For then,  $v_{k+1}$  may be computed with much higher relative accuracy, and the iterates lie substantially closer to the null-space of  $A$ . Picking  $y_{k+1}$  as the previously generated  $u_k$  appears to be effective in practice.

### Notes and References for Subsection 5.4.1

The idea of projecting the conjugate gradient method into the null-space of  $A$  dates back at least to Polyak (1969). The general formulation, involving (5.4.5), is due to Coleman (1994), although he unnecessarily requires that  $M$ , rather than simply  $N^T M N$ , be positive definite. Precautions, such as the alternative (5.4.7) and the use of iterative refinement, needed to ensure that the method behaves well in practice are given by Gould, Hribar, and Nocedal (1998).

*Iterative refinement* is a method by which the accuracy of the computed solution of any linear system  $Ax = b$  may be improved. Simply, if  $x^C$  is an estimate of the system, then the refined estimate  $x^C + \Delta x^C$ , where  $\Delta x^C$  is the computed solution of  $A\Delta x^C = b - Ax^C$ , is often a better estimate. Clearly, such a procedure may be applied iteratively, the same method being applied to the residual  $b - A(x^C + \Delta x^C)$ . While early variants suggested that the residual  $b - Ax^C$  should be computed in extended-precision arithmetic, more recently it has been recognized that iterative refinement is often beneficial even when extra precision is not used. See Wilkinson (1965), Golub and Van Loan (1989), or Higham (1996) for details.

## Part II

---

# Trust-Region Methods for Unconstrained Optimization

In this, the first formal part of our description of trust-region methods, we introduce the reader to the central ideas that lie behind all trust-region methods. The basic problem we consider is that of minimizing an unconstrained objective function without further restrictions on the values its variables may take.

---

# Chapter 6

---

## Global Convergence of the Basic Algorithm

---

### 6.1 The Basic Trust-Region Algorithm

The purpose of this chapter is to provide an in-depth introduction to trust-region methods for the solution of unconstrained problems. More precisely, we will consider a class of algorithms for finding a local solution of the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x), \quad (6.1.1)$$

where  $f(x)$ , the objective function, is a real-valued twice-continuously differentiable function, which we assume is bounded below. The algorithms of the class, according to our discussion of Chapter 1, are iterative numerical procedures in which the objective function  $f(x)$  is approximated in a suitable neighbourhood of the current iterate (the trust region) by a model which is easier to handle than  $f(x)$ . In this chapter and its successors, we will describe the essence of a rich body of convergence results that may be derived for this class. We will do this by making all assumptions we need to keep the framework simple and, we hope, intuitive. We shall see later how several of these assumptions may be relaxed, for greater generality, once we have established the basic theory in our simple framework.

We first introduce some notation. We consider an iterative technique producing a sequence of *iterates*  $\{x_k\}$  which we hope converges to a *first-order critical point* of problem (6.1.1), as we described in Section 3.2.1. If possible, such a first-order critical point should also be a (local) minimizer of the objective function.

Recalling the broad description of page 6, a basic trust-region algorithm works as follows. At each iterate  $x_k$ , we first define a model  $m_k(x)$  whose purpose is to approximate the objective function within a suitable neighbourhood of  $x_k$ , which we refer to as the trust region. The *trust region* is the set of all points

$$\mathcal{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k\}, \quad (6.1.2)$$

where  $\Delta_k$  is called the *trust-region radius*, and where  $\|\cdot\|_k$  is an iteration-dependent norm. Given this model and its trust region, we next seek a *trial step*  $s_k$  to a *trial point*  $x_k + s_k$  with the aim of reducing the model while satisfying the bound  $\|s_k\|_k \leq \Delta_k$ . We ask that the reduction in the model at the trial point bears some relationship to the value of  $\nabla_x f(x_k)$  (we shall shortly see why this is necessary, and how we can ensure that it is so) and say that any step which achieves this gives a *sufficient* reduction in the model. Having determined the trial step, the objective function is now computed at  $x_k + s_k$  and compared to the value predicted by the model at this point, that is,  $m_k(x_k + s_k)$ . If the sufficient reduction predicted by the model is realized by the objective function, the trial point is accepted as the next iterate and the trust region is expanded or kept the same. If the model reduction turns out to be a poor predictor of the actual behaviour of the objective function, the trial point is rejected and the trust region is contracted, with the hope that the model provides a better prediction in the smaller region. More formally, our basic trust-region algorithm, which we call Algorithm BTR, may be described as follows.

**Algorithm 6.1.1: Basic trust-region algorithm (BTR)**

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy

$$0 < \eta_1 \leq \eta_2 < 1 \quad \text{and} \quad 0 < \gamma_1 \leq \gamma_2 < 1. \quad (6.1.3)$$

Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m_k$  in  $\mathcal{B}_k$ .

**Step 2: Step calculation.** Compute a step  $s_k$  that “sufficiently reduces the model”  $m_k$  and such that  $x_k + s_k \in \mathcal{B}_k$ .

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \quad (6.1.4)$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (6.1.5)$$

Increment  $k$  by 1 and go to Step 1.

In this description, reasonable values for the constants of (6.1.3) are, for instance,

$$\eta_1 = 0.01, \quad \eta_2 = 0.9, \quad \text{and} \quad \gamma_1 = \gamma_2 = \frac{1}{2}, \quad (6.1.6)$$

but other values may be preferable. We pursue this discussion in Section 17.1. Iterations for which  $\rho_k \geq \eta_1$ , and thus for which  $x_{k+1} = x_k + s_k$ , are called *successful iterations*, and we denote the set of their indices by the symbol  $\mathcal{S}$ , that is,

$$\mathcal{S} = \{k \geq 0 \mid \rho_k \geq \eta_1\}.$$

Similarly, we also define

$$\mathcal{V} = \{k \geq 0 \mid \rho_k \geq \eta_2\},$$

the set of iterations that are *very successful*. Note that  $\mathcal{V} \subseteq \mathcal{S}$ . Iterations whose index does not belong to  $\mathcal{S}$  are said to be *unsuccessful*.

This framework is, for now, quite general and leaves many details unspecified. For instance, we do not prescribe which model  $m_k$  to use in Step 1. In practice, one often chooses a quadratic model of the form

$$m_k(x_k + s) = m_k(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle, \quad (6.1.7)$$

where

$$m_k(x_k) = f(x_k) \quad \text{and} \quad g_k = \nabla_x f(x_k)$$

and where  $H_k$  is a symmetric approximation to  $\nabla_{xx} f(x_k)$ . If  $H_k \neq 0$ , we say that (6.1.7) is a second-order model. More generally, we say that a model  $m_k$  is of *order*  $\ell$  whenever its highest nonidentically vanishing derivative is the  $\ell$ th derivative. The precise way in which the step  $s_k$  is computed or what we mean by “sufficient reduction” are not yet described. It is the purpose of the next two sections to clarify some of these issues, as well as to say what we require of problem (6.1.1) itself.

As specified, Algorithm BTR does not include a stopping test. This is, of course, unrealistic from the practical point of view but reasonable when analysing the convergence properties of the algorithm; we thus assume that an infinite sequence of iterates  $\{x_k\}$  is generated. If Algorithm BTR is implemented as a computer program, it will presumably be stopped as soon as the iterate  $x_k$  is judged to be “critical” enough.<sup>48</sup> Of course this requires that we know how to measure the criticality of  $x_k$ : in our unconstrained context, the simplest such measure is the norm of the gradient<sup>49</sup> of the objective function at  $x_k$ ,  $\|\nabla_x f(x_k)\|$ . We shall consider this issue further in Section 17.4.3. The reader should also note that the trust-region management as specified by Step 4 leaves considerable freedom in the way  $\Delta_{k+1}$  is selected. We will return to this aspect of the algorithm in Sections 10.5.2 and 17.1.

An important practical aspect of any trust-region method is the shape of the trust region itself, which is determined by the norm used in (6.1.2). Ideally, this shape should

---

<sup>48</sup>Most programs would also set a maximum on the number of iterations allowed.

<sup>49</sup>Better measures are possible that take the problem scaling into account; see Section 8.1.4.

reflect the region where we believe the model approximates the objective function well, that is, where the ratio

$$\rho(s) = \frac{f(x_k) - f(x_k + s)}{m_k(x_k) - m_k(x_k + s)},$$

which is just (6.1.4) considered as a function of a trial step  $s$ , is close to 1. Unfortunately, this shape strongly depends on the nonlinearity of the objective function, as we illustrate in Figure 6.1.1. In this figure, we first consider the famous Rosenbrock's function (see Rosenbrock, 1960)

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

whose contour lines are shown in the top picture of the figure. We then select two different  $x_k$  (indicated by small circles), construct quadratic models of the objective (based on the second-order Taylor series of  $f$  at  $x_k$ ), and compute the function  $\rho(s)$ . We finally plot the contour lines of this function  $\rho$  for values 0 (the limit of the region where  $f(x_k + s) \leq f(x_k)$ ), 0.25, and 0.75 (reasonable values for the constants  $\eta_1$  and  $\eta_2$ ; see (6.1.6)). The regions for which  $\rho(s) \geq 0.75$  are of course included in those for which  $\rho(s) \geq 0.25$ , which are themselves included in the regions where  $\rho(s) \geq 0$ . These two contour plots are shown in the bottom pictures of the figure. The insides of those

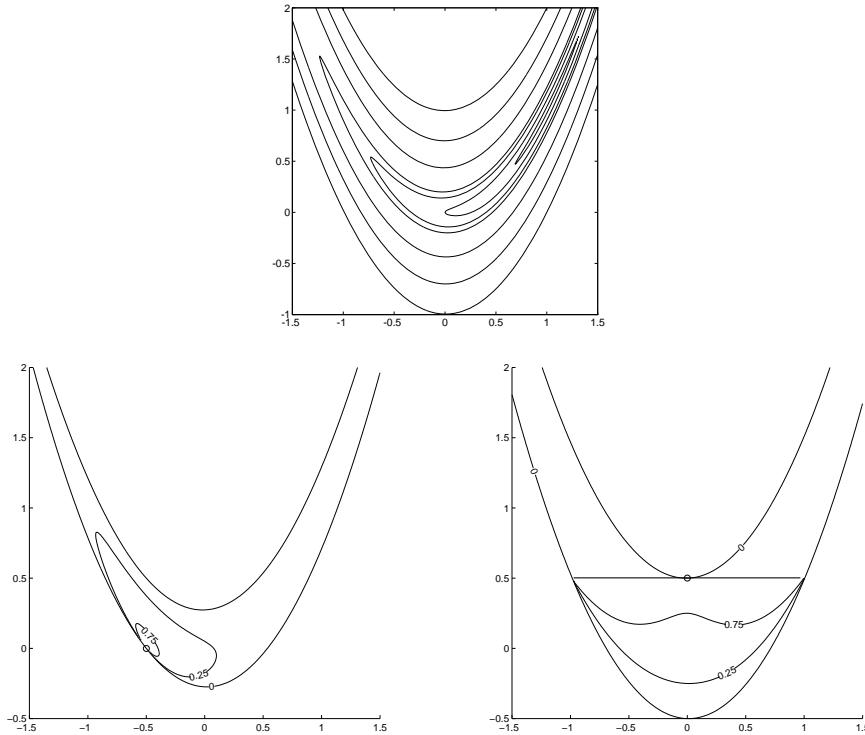


Figure 6.1.1: The ideal trust-region shape for Rosenbrock's function and different choices of  $x_k$ .

regions give the ideal shape of the trust region, but we see that this shape may be extremely complicated (it may even consist of several disconnected components).

We therefore see that it is preferable to specify the shape of the trust region in a problem- and iteration-dependent manner, which is why we choose to denote the norm by  $\|\cdot\|_k$  in (6.1.2). Figure 6.1.2 illustrates the different trust-region shapes obtained for the usual  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms using the same trust-region radius.

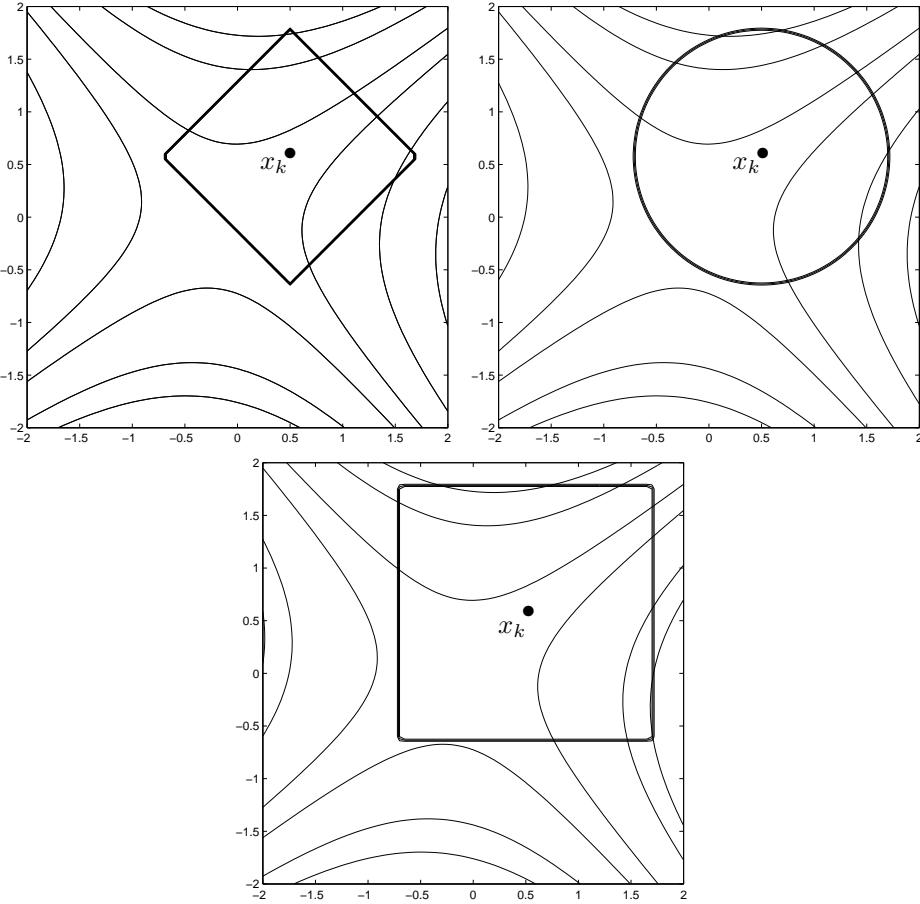


Figure 6.1.2: The trust-region shape when the trust region is defined using the  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms.

One could also think of other norms, for instance when the problem at hand is a discretization of an infinite-dimensional problem,<sup>50</sup> in which case the inner product, and thus the subordinate norm, should reflect the structure of the infinite-dimensional space. Another important case where we may wish to change our norms in (6.1.2) is when the problem's variables are badly scaled. We will return to the important problem of choosing the norm  $\|\cdot\|_k$  and the shape of the trust region in Section 6.7.

---

<sup>50</sup>See Section 8.3.

## Notes and References for Section 6.1

Although the basic algorithm described in this section is, by far, the most common framework for trust-region methods and the basis for many variants of interest,<sup>51</sup> it is not the only possible one. For instance, Shultz, Schnabel, and Byrd (1985) propose a more general setting, where the computation of the step and of a suitable trust-region radius are simultaneous, by contrast to Algorithm BTR, where they are strictly sequential. See Section 10.5.1 and the associated notes for the discussion of a potential advantage of this technique. Another “nonstandard” trust-region method is discussed in Osborne (1998). In this algorithm, iterations at which the ratio of achieved versus predicted reduction is “too good”, in the sense that  $\rho_k > \eta_2$ , are defined to be unsuccessful (in particular, and somewhat surprisingly, a perfect model fit, resulting in  $\rho_k = 1$ , leads to an unsuccessful iteration), and the trust-region radius is then decreased.

The Harwell Subroutine Library (2000) code **VA06** written by Powell (1970a) appears to be one of the first widely available codes for trust-region methods. It corresponds to the unconstrained optimization method of Powell (1970d). Since then, a number of other packages have used the trust-region technique. Gay (1983) describes trust-region subroutines based on Gay (1982) in the framework of the PORT 3 package (see Fox, Hall, and Schryer, 1978). The **UNCMIN** package of Schnabel, Koontz, and Weiss (1985) contains trust-region methods for unconstrained problems, along with linesearch ones. Its underlying algorithm and code are explained in Dennis and Schnabel (1983). **VE08** by Toint (1983b), was the first trust-region method for large-scale unconstrained problems. It uses the all-pervasive partially separable<sup>52</sup> structure of such problems to achieve numerical efficiency. More recent packages containing trust-region methods for unconstrained optimization include **LANCELOT** by Conn, Gould, and Toint (1992b); **STENMIN** by Bouaricha (1997), which uses a high-order model of the objective function; and the recent **TRON** (trust-region optimization with Newton’s method) code described by Lin and Moré (1999a). Several application-dependent packages, such as **HieLoW** by Bierlaire (1994) or **ORDPACK** by Boggs et al. (1989), are also based on the trust-region idea, but it is impossible to review them all here. The reader is referred to Moré and Wright (1993) for additional information.

We should also mention that there is another class of methods, very close in spirit to trust-region methods, called *proximal point methods*, which were introduced by Martinet (1970) and popularized by Rockafellar (1976a, 1976b). These methods are typified by an iteration of the form

$$x_k = \arg \min_{x \in \mathbb{R}^n} \left[ f(x) + \frac{1}{\mu_k} \text{distance}(x, x_{k-1}) \right].$$

Various concepts of distance can be used in this definition, such as the simple Euclidean norm or, for constrained problems, the Bregman distance (see Bregman, 1967; Iusem, 1991; or Eckstein, 1993) or the class of “ $\varphi$ -divergences”, including the Kullback–Leibler relative entropy distance functional (see Teboulle, 1997). The connection with trust-region methods is that one replaces the constraint that the next iterate must be in a well-defined neighbourhood of the current point (the trust region) by a term in the objective function that penalizes the “length” of the step. The strength of this penalty is governed by the iteration-dependent parameter  $\mu_k$ : small values of  $\mu_k$  correspond to a strong penalty, which itself can be linked to a small trust region and thus to a small value of the trust-region radius. Note the similarity

---

<sup>51</sup>See, for instance, Chapter 10.

<sup>52</sup>See Section 10.2.

with the ideas of Levenberg (1944), Morrison (1960), and Marquardt (1963).

Proximal point methods are the subject of very active research, in particular for the solution of nonsmooth optimization problems. We refer the reader to Bonnans et al. (1995), Kiwiel (1996), or Zhu (1996) for more information. See also Jonsson and Larsson (1990) for an application in structural optimization. Interestingly, and somewhat surprisingly given the similarity that we have just pointed out, it seems that analysis has been confined so far to convex problems, although the implicit trust-region scheme discussed in Burke and Weigmann (1997) appears to be a first step in the direction of more general cases. Because of our strong interest in nonconvex problems, we will not explore the connection between trust-region and proximal point methods further.

## 6.2 Assumptions

In order to highlight the assumptions necessary for our convergence theory, we group them all in this section, distinguishing between the assumptions on the optimization problem and those on our algorithm.

### 6.2.1 Assumptions on the Problem

For clarity, we repeat here our precise assumptions on the problem (6.1.1), beyond AF.1.

**AF.2**  $f(x)$  is bounded below on  $\mathbb{R}^n$ ; that is, there exists a constant<sup>53</sup>  $\kappa_{\text{lbf}}$  such that, for all  $x \in \mathbb{R}^n$ ,

$$f(x) \geq \kappa_{\text{lbf}}.$$

**AF.3** The Hessian of the objective function is uniformly bounded; that is, there exists a positive constant<sup>54</sup>  $\kappa_{\text{ufh}}$  such that, for all  $x \in \mathbb{R}^n$ ,

$$\|\nabla_{xx} f(x)\| \leq \kappa_{\text{ufh}}.$$

We immediately note that this latter assumption is often too strong. In fact, we normally need boundedness of  $\|\nabla_{xx} f(x)\|$  for values of  $x$  that lie between two successive iterates of the basic algorithm. This weaker requirement is thus automatically satisfied if the iterates remain in a bounded subset of  $\mathbb{R}^n$ , such as, for instance, the level set  $\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$  for some  $x_0$ . Note that we may suppose, without loss of generality, that

$$\kappa_{\text{ufh}} \geq 1.$$

---

<sup>53</sup>“lbf” stands for “lower bound on the objective function”.

<sup>54</sup>“ufh” stands for “upper bound on the objective function’s Hessian”.

### 6.2.2 Assumptions on the Algorithm

As our purpose is to simplify exposition as much as possible without hiding the main ideas, we will assume that the model  $m_k$  chosen at iteration  $k$  to represent the objective function within the trust region  $\mathcal{B}_k$  is a good first-order smooth approximation of the objective. Consequently, we require the following assumptions.

**AM.1** For all  $k$ , the model  $m_k$  is twice differentiable on  $\mathcal{B}_k$ .

**AM.2** The values of the objective function and of the model coincide at the current iterate; that is, for all  $k$ ,

$$m_k(x_k) = f(x_k). \quad (6.2.1)$$

**AM.3** The gradient of the model at  $x_k$  is equal to the gradient of the objective function; that is, for all  $k$ ,

$$g_k \stackrel{\text{def}}{=} \nabla_x m_k(x_k) = \nabla_x f(x_k). \quad (6.2.2)$$

**AM.4** The Hessian of the model remains bounded within the trust region; that is,

$$\|\nabla_{xx} m_k(x)\| \leq \kappa_{\text{umh}} - 1 \quad \text{for all } x \in \mathcal{B}_k, \quad (6.2.3)$$

for all  $k$ , where  $\kappa_{\text{umh}} \geq 1$  is a constant<sup>55</sup> independent of  $k$ .

We immediately note that these assumptions are fulfilled if we consider a simple but very important trust-region variant of Newton's method over a closed bounded domain, that is, if we choose  $m_k$  such that (6.2.1) and (6.2.2) hold, as well as

$$\nabla_{xx} m_k(x_k + s) = \nabla_{xx} f(x_k), \quad (6.2.4)$$

for all  $s$  such that  $x_k + s \in \mathcal{B}_k$ . If we assume that the objective function is twice differentiable (i.e., AF.1), equation (6.2.3) then results from (6.2.4) and AF.3. The main reason to develop the theory for the more general case is to clarify the relationship between the objective function and its model. This would be less apparent if (6.2.4) were assumed throughout. Moreover, the wider applicability of the more general theory justifies the additional complexity. We also note that AM.1–AM.4 allow for the case where the complete objective function cannot be approximated by a model. For instance, we may consider the case in which the objective function is of the form

$$f(x) = f_0(x, y(x)), \quad (6.2.5)$$

---

<sup>55</sup>“umh” stands for “upper bound on the model's Hessian”. We use  $\kappa_{\text{umh}} - 1$  in (6.2.3) instead of the more natural  $\kappa_{\text{umh}}$  purely for notational convenience. See, for instance, (6.3.1). Also observe that we have chosen names for our assumptions in accordance with our policy stated on page xviii.

where  $f_0(x, y)$  is a function from  $\mathbb{R}^n \times \mathbb{R}^p$  into  $\mathbb{R}$  which is relatively simple to compute, but  $y(x)$  is complicated and costly from  $\mathbb{R}^n$  to  $\mathbb{R}^p$ . For instance,  $f_0$  might be a simple performance criterion for a system whose state  $y(x)$  can only be calculated by using a complicated computational procedure, such as solving a partial differential equation or running a dedicated simulation. In this case, it may be advisable to construct a model  $m_k^y(x)$  of  $y(x)$  in the neighbourhood of  $x_k$  and then define

$$m_k(x) = f_0(x, m_k^y(x)). \quad (6.2.6)$$

The conditions AM.1–AM.4 of the model  $m_k$  may then be reformulated<sup>56</sup> as conditions on  $m_k^y$  and  $f_0$ . Our choice to keep the model as general as possible subject to AM.1–AM.4 thus ensures that models of the type (6.2.6) do not need special treatment as far as the convergence theory is concerned.

We complete our assumptions on the algorithm by specifying the relation between the various norms  $\|\cdot\|_k$  defining the trust-region shape in (6.1.2) (p. 115).

**AN.1** There exists a constant<sup>57</sup>  $\kappa_{\text{une}} \geq 1$  such that, for all  $k$ ,

$$\frac{1}{\kappa_{\text{une}}} \|x\|_k \leq \|x\| \leq \kappa_{\text{une}} \|x\|_k$$

for all  $x \in \mathbb{R}^n$ .

We then say that the norms  $\|\cdot\|_k$  are *uniformly equivalent* to the Euclidean  $\ell_2$  norm. This merely says that the trust region may not asymptotically “completely flatten” in any direction, as  $k$  increases. We return to a detailed discussion of this assumption in Sections 6.7 and 8.2.

## 6.3 The Cauchy Point and the Model Decrease

A crucial point in our algorithm is the determination, at Step 2, of the step  $s_k$ , which “sufficiently reduces” the model within the trust region. In this subsection, we derive a formal definition of this property from a very simple computational technique.

### 6.3.1 The Cauchy Arc

One of the simplest possible strategies for reducing the model within the trust region is to examine the behaviour of the model along the steepest descent  $-g_k$  within the trust region  $\mathcal{B}_k$ . As its name suggests, it is along this direction that the model locally decreases at the fastest rate, and we may therefore anticipate obtaining a good model reduction if we move as far as we can in this direction while the model continues to

---

<sup>56</sup>This reformulation and the general context of (6.2.5)–(6.2.6) are discussed in more detail in Section 8.5.

<sup>57</sup>“une” stand for “uniform norm equivalence”. Note that this assumption’s name (see page xviii) indicates it is concerned with norms.

decrease. In other words, we would like to calculate a minimum of the model along the *Cauchy arc* defined by

$$x_k^C(t) \stackrel{\text{def}}{=} \{x \mid x = x_k - tg_k, t \geq 0 \text{ and } x \in \mathcal{B}_k\}.$$

The Cauchy arc is illustrated in Figure 6.3.1. In the first figure, the model (whose contour lines are shown) is indefinite, while it is nicely convex in the second. Note that  $x_k^C(t) = x_k$  for all  $t \geq 0$  whenever  $g_k = 0$ .

The developments that follow naturally consider the curvature of the model, mainly along the Cauchy arc. In this context, we introduce

$$\beta_k \stackrel{\text{def}}{=} 1 + \max_{x \in \mathcal{B}_k} \|\nabla_{xx} m_k(x)\| \quad (6.3.1)$$

as an upper bound on this curvature.<sup>58</sup> This definition and AM.4 also imply that

$$\beta_k \leq \kappa_{\text{umh}} \quad (6.3.2)$$

for all  $k$ .

### 6.3.2 The Cauchy Point for Quadratic Models

If we assume that our model is quadratic, that is, that it is of the form (6.1.7) (p. 117), then it is possible to minimize it exactly on the Cauchy arc. The resulting point,

$$x_k^C = x_k - t_k^C g_k = \arg \min_{\substack{t \geq 0 \\ x_k - tg_k \in \mathcal{B}_k}} m_k(x_k - tg_k), \quad (6.3.3)$$

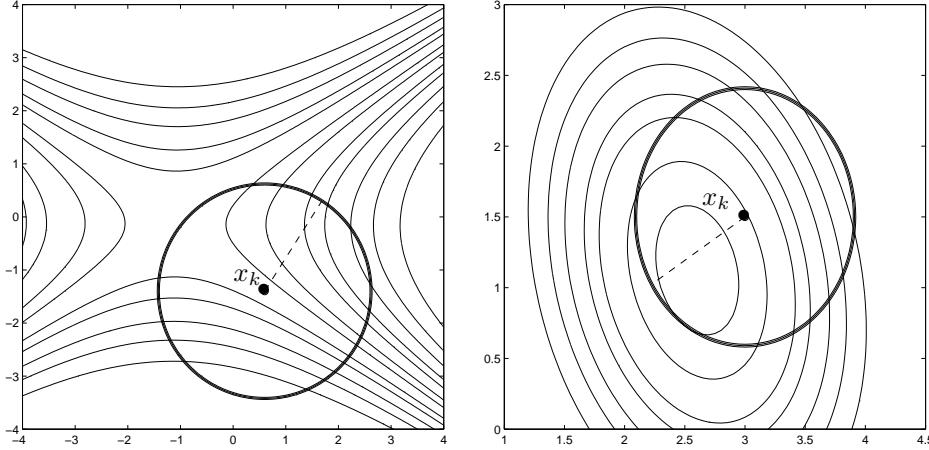


Figure 6.3.1: The trust region and the Cauchy arc for indefinite and convex models. The trust-region boundary is indicated by a thick circle, and the Cauchy arc by a dotted line.

<sup>58</sup>The constant 1 in the definition of  $\beta_k$  is arbitrary. Any strictly positive value would also be acceptable.

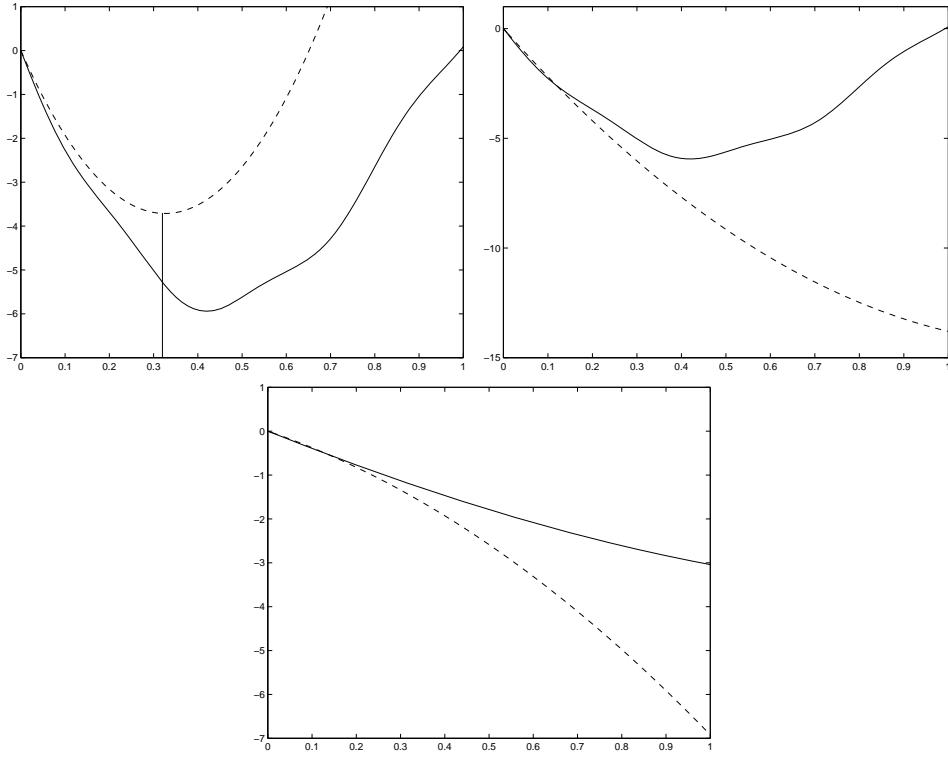


Figure 6.3.2: The three cases that may occur when determining the Cauchy point, depending on the model's curvature along the Cauchy arc. The solid and dashed lines are, respectively, plots of  $f(x_k - t\Delta_k g_k / \|g_k\|_k)$  and  $m_k(x_k - t\Delta_k g_k / \|g_k\|_k)$  as functions of  $t$ . In the top left figure, the model is convex and its unconstrained minimizer lies within the trust region; thus it provides the Cauchy point. The model in the top right figure is also convex, but now its unconstrained minimizer lies outside the trust region. For this example, and for the nonconvex model illustrated in the bottom figure, the Cauchy point lies on the trust-region boundary.

that is, the (unique) minimizer of the model along the Cauchy arc, is called the *Cauchy point*. This point will play a central role in the convergence theory, principally because it allows a suitable characterization of the decrease in the model which is reasonable to expect for a single iteration.

If this definition is applied, three cases may arise, depending on the curvature of the model along the Cauchy arc. All three cases are illustrated in Figure 6.3.2.

The analysis of these three cases leads to the following result.

**Theorem 6.3.1** If the model is given by (6.1.7) (p. 117), and if we define the Cauchy point by (6.3.3), we have that

$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \nu_k^C \Delta_k \right], \quad (6.3.4)$$

where  $\beta_k$  is defined by (6.3.1) and where

$$\nu_k^C = \frac{\|g_k\|}{\|g_k\|_k}. \quad (6.3.5)$$

**Proof.** We first note that, for all  $t \geq 0$ ,

$$m_k(x_k - tg_k) = m_k(x_k) - t\|g_k\|^2 + \frac{1}{2}t^2\langle g_k, H_k g_k \rangle. \quad (6.3.6)$$

We now consider the case in which the curvature of the model along the steepest descent is positive, that is, when

$$\langle g_k, H_k g_k \rangle > 0, \quad (6.3.7)$$

and compute the value of the parameter  $t$  at which the unique minimum of (6.3.6) is attained. Let us denote this optimal parameter by  $t_k^*$ . Differentiating (6.3.6) with respect to  $t$  and equating the result to zero, we obtain that

$$0 = \|g_k\|^2 - t_k^* \langle g_k, H_k g_k \rangle,$$

which immediately gives that

$$t_k^* = \frac{\|g_k\|^2}{\langle g_k, H_k g_k \rangle}. \quad (6.3.8)$$

Two subcases may then occur. The first is when this minimizer lies within the trust region, that is, when  $t_k^* \|g_k\|_k \leq \Delta_k$ . This corresponds to the top left picture in Figure 6.3.2. Then  $t_k^C = t_k^*$  and we may replace this expression in the model decrease (6.3.6), which allows us to deduce that

$$m_k(x_k) - m_k(x_k^C) = \frac{\|g_k\|^4}{\langle g_k, H_k g_k \rangle} - \frac{1}{2} \frac{\|g_k\|^4}{\langle g_k, H_k g_k \rangle} = \frac{1}{2} \frac{\|g_k\|^4}{\langle g_k, H_k g_k \rangle} \geq \frac{\|g_k\|^2}{2\beta_k}, \quad (6.3.9)$$

where we used the fact that  $|\langle g_k, H_k g_k \rangle| \leq \|g_k\|^2 \beta_k$  because of the Cauchy–Schwarz inequality and (6.3.1). If, on the other hand,

$$t_k^* \|g_k\|_k > \Delta_k, \quad (6.3.10)$$

as illustrated by the top right frame of Figure 6.3.2, then the line minimum is outside the trust region and we have that

$$t_k^C \|g_k\|_k = \Delta_k. \quad (6.3.11)$$

Combining (6.3.8), (6.3.10), and (6.3.11), we see that

$$\langle g_k, H_k g_k \rangle < \frac{\|g_k\|^2}{t_k^C}.$$

Substituting this last inequality in (6.3.6) and using (6.3.11), we obtain that

$$\begin{aligned} m_k(x_k) - m_k(x_k^C) &= t_k^C \|g_k\|^2 - \frac{1}{2}[t_k^C]^2 \langle g_k, H_k g_k \rangle \\ &> \nu_k^C \|g_k\| \Delta_k - \frac{1}{2} \nu_k^C \|g_k\| \Delta_k \\ &= \frac{1}{2} \nu_k^C \|g_k\| \Delta_k, \end{aligned} \quad (6.3.12)$$

where  $\nu_k^C$  is given by (6.3.5).

Finally, we consider the case where the curvature of the model along the steepest-descent direction is negative, that is, when (6.3.7) is violated. This last case is shown in the bottom picture of Figure 6.3.2. We then obtain from (6.3.6) that

$$m_k(x_k - tg_k) = m_k(x_k) - t\|g_k\|^2 + \frac{1}{2}t^2 \langle g_k, H_k g_k \rangle \leq m_k(x_k) - t\|g_k\|^2 \quad (6.3.13)$$

for all  $t \geq 0$ . In that case, it is easy to see that the Cauchy point must lie on the boundary of the trust region, and thus that (6.3.11) holds. Combining this equality and (6.3.13), we deduce that

$$m_k(x_k) - m_k(x_k^C) \geq \|g_k\|^2 \frac{\Delta_k}{\|g_k\|_k} = \nu_k^C \|g_k\| \Delta_k \geq \frac{1}{2} \nu_k^C \|g_k\| \Delta_k. \quad (6.3.14)$$

We may then conclude our proof by noting that (6.3.9), (6.3.12), and (6.3.14) imply that (6.3.4) holds.  $\square$

Theorem 6.3.1 is a special case of a general result for the minimization of a quadratic function

$$q(s) = f + \langle g, s \rangle + \frac{1}{2} \langle s, H s \rangle \quad (6.3.15)$$

over all points along the arc  $s = -tg$  within the region

$$\|s\|_a \leq \Delta, \quad (6.3.16)$$

where  $\|\cdot\|_a$  is a given, arbitrary norm. Since we shall use the general result in later sections of our book, we state it formally here.

**Corollary 6.3.2** Suppose that  $s^C$  is the minimizer of the quadratic function (6.3.15) within the trust region (6.3.16) for all points lying along the arc  $s(t) = -tg$ . Then we have that

$$q(0) - q(s^C) \geq \frac{1}{2} \|g\| \min \left[ \frac{\|g\|}{1 + \|H\|}, \nu^C \Delta \right],$$

where

$$\nu^C = \frac{\|g\|}{\|g\|_a}.$$

**Proof.** This result follows directly from Theorem 6.3.1 if we make the connections  $q(s) = m_k(x_k + s)$ ,  $f = m_k(x_k)$ ,  $g = g(x_k)$ ,  $H = H_k$ ,  $\Delta = \Delta_k$ , and  $\|\cdot\|_a = \|\cdot\|_k$ , and replace  $\beta_k$  by the quantity of which it is an upper bound.  $\square$

### 6.3.3 The Approximate Cauchy Point

If we continue to allow our model to be general, an exact minimizer may be more difficult to compute, and we may instead use a simple backtracking strategy to find a point that provides a good model reduction. More precisely, we could determine the smallest nonnegative integer  $j = j_c$  such that the point

$$x_k(j) \stackrel{\text{def}}{=} x_k - \kappa_{\text{bck}}^j \frac{\Delta_k}{\|g_k\|_k} g_k \quad (6.3.17)$$

satisfies the conditions

$$m_k(x_k(j)) \leq m_k(x_k) + \kappa_{\text{ubs}} \langle g_k, x_k(j) - x_k \rangle, \quad (6.3.18)$$

where  $\kappa_{\text{bck}} \in (0, 1)$  and  $\kappa_{\text{ubs}} \in (0, \frac{1}{2})$  are given constants.<sup>59</sup> This is the application, in our context, of the well-known Armijo linesearch (see, for instance, Dennis and Schnabel, 1983, pages 117ff.). It is also referred to as a *backtracking* technique because the point resulting from the procedure is obtained by moving backwards from  $x_k(0)$  to the current iterate  $x_k$ . We may then define the *approximate Cauchy point* as

$$x_k^{\text{AC}} \stackrel{\text{def}}{=} x_k(j_c).$$

We illustrate the Armijo search for the approximate Cauchy point in Figure 6.3.3.

We now consider what model reduction can be ensured when moving from the current iterate  $x_k$  to the approximate Cauchy point  $x_k^{\text{AC}}$ . This is the object of the next proposition.

**Theorem 6.3.3** Suppose that AM.1 holds. Then the approximate Cauchy point  $x_k^{\text{AC}}$  is well defined in the sense that  $j_c$  is finite. Moreover,

$$m_k(x_k) - m_k(x_k^{\text{AC}}) \geq \kappa_{\text{dcp}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \nu_k^{\text{C}} \Delta_k \right], \quad (6.3.19)$$

where  $\kappa_{\text{dcp}} \in (0, 1)$  is a constant<sup>60</sup> independent of  $k$ , where  $\beta_k$  is defined by (6.3.1), and where  $\nu_k^{\text{C}}$  is defined by (6.3.5).

**Proof.** We first consider the case where condition (6.3.18) is violated for some  $j$ , and therefore we have that

$$m_k(x_k - t_j g_k) > m_k(x_k) - \kappa_{\text{ubs}} t_j \|g_k\|^2, \quad (6.3.20)$$

where we have defined

$$t_j = \kappa_{\text{bck}}^j \Delta_k / \|g_k\|_k.$$

<sup>59</sup>“bck” stands for “backtracking” and “ubs” for “upper bound on the slope”.

<sup>60</sup>“dcp” stands for “decrease at the Cauchy point”.

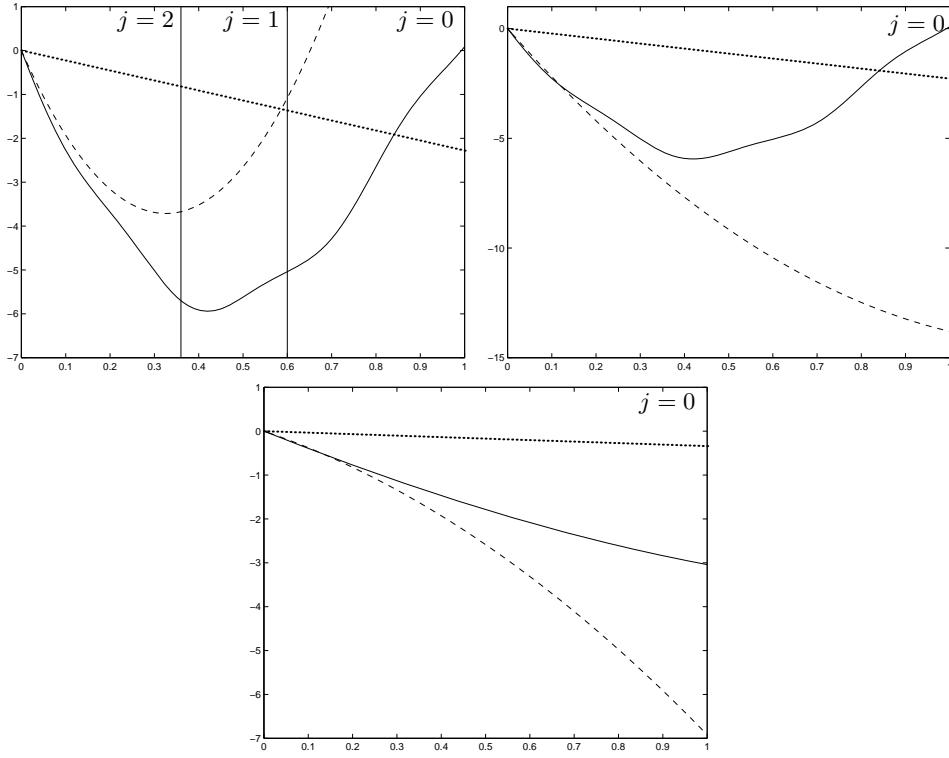


Figure 6.3.3: The three cases that may occur when determining the approximate Cauchy point by backtracking, depending on the model's curvature along the Cauchy arc. The solid, dashed, and dotted lines are, respectively, plots of the objective function,  $f(x_k - t\Delta_k g_k / \|g_k\|_k)$ , the model,  $m_k(x_k - t\Delta_k g_k / \|g_k\|_k)$ , and the linear approximation defined by the right-hand side of condition (6.3.18),  $m_k(x_k) - t\kappa_{\text{ubs}}\Delta_k\|g_k\|^2/\|g_k\|_k$ , as functions of  $t$ . The backtracking parameters are chosen as  $\kappa_{\text{bck}} = 0.6$  and  $\kappa_{\text{ubs}} = 0.01$ . Notice in the top left figure that the model's curvature is positive and that (6.3.18) is violated for  $j = 0$  ( $t = 1$ ) and, barely, for  $j = 1$  ( $t = 0.6$ ) but is finally satisfied for  $j_c = 2$  ( $t = 0.36$ ), which we indicate by the vertical lines corresponding to successive values of  $\kappa_{\text{bck}}^j$ . In the upper right and bottom figures, the model's curvature is too small to require backtracking. In these cases, the approximate Cauchy point lies at the boundary of the trust region ( $j_c = 0$ ).

This situation is illustrated in the first frame of Figure 6.3.3. We may now use the mean value theorem for the left-hand side of (6.3.20) with a  $\xi_j$  belonging to the segment  $[x_k, x_k - t_j g_k]$ , and deduce that

$$m_k(x_k) - t_j \|g_k\|^2 + \frac{1}{2}t_j^2 \langle g_k, \nabla_{xx} m_k(\xi_j) g_k \rangle > m_k(x_k) - \kappa_{\text{ubs}} t_j \|g_k\|^2. \quad (6.3.21)$$

On the other hand, taking into account that  $\|\nabla_{xx} m_k(\xi_j)\| \leq \beta_k$  because of (6.3.1)

and the inclusion  $\xi_j \in \mathcal{B}_k$ , we obtain from the Cauchy–Schwarz inequality that

$$m_k(x_k) - t_j \|g_k\|^2 + \frac{1}{2} t_j^2 \langle g_k, \nabla_{xx} m_k(\xi_j) g_k \rangle \leq m_k(x_k) - t_j \|g_k\|^2 + \frac{1}{2} t_j^2 \beta_k \|g_k\|^2. \quad (6.3.22)$$

Now combining (6.3.21) and (6.3.22), we see that

$$t_j > \frac{2(1 - \kappa_{\text{ubs}})}{\beta_k}.$$

As a consequence and because  $\kappa_{\text{bck}} < 1$ , there must be a first finite  $j_c$  such that

$$\frac{\kappa_{\text{bck}}^{j_c} \Delta_k}{\|g_k\|_k} < \frac{2(1 - \kappa_{\text{ubs}})}{\beta_k}, \quad (6.3.23)$$

for which we have that (6.3.18) holds. The approximate Cauchy point  $x_k^{\text{AC}}$  is thus well defined, and we obtain from (6.3.17) and (6.3.18) that

$$m_k(x_k) - m_k(x_k^{\text{AC}}) \geq -\kappa_{\text{ubs}} \langle g_k, x_k(j_c) - x_k \rangle = \kappa_{\text{ubs}} \kappa_{\text{bck}}^{j_c} \nu_k^{\text{C}} \Delta_k \|g_k\|, \quad (6.3.24)$$

where  $\nu_k^{\text{C}}$  is defined by (6.3.5).

Now, if  $j_c \geq 1$  (that is,  $x_k^{\text{AC}}$  lies in the interior of the trust region; see the left part of Figure 6.3.3) and since it is the smallest  $j$  that ensures (6.3.23), we may deduce that

$$\kappa_{\text{bck}}^{j_c} = \kappa_{\text{bck}} \kappa_{\text{bck}}^{j_c-1} \geq 2\kappa_{\text{bck}}(1 - \kappa_{\text{ubs}}) \frac{\|g_k\|_k}{\beta_k \Delta_k},$$

which, together with (6.3.24) and (6.3.5), gives that

$$m_k(x_k) - m_k(x_k^{\text{AC}}) \geq 2\kappa_{\text{bck}} \kappa_{\text{ubs}} (1 - \kappa_{\text{ubs}}) \frac{\|g_k\|^2}{\beta_k}. \quad (6.3.25)$$

If, on the other hand,  $j_c = 0$ , that is, if  $x_k^{\text{AC}}$  lies on the boundary of the trust region as illustrated by the second and third frames of Figure 6.3.3, we immediately deduce from (6.3.24) that

$$m_k(x_k) - m_k(x_k^{\text{AC}}) \geq \kappa_{\text{ubs}} \nu_k^{\text{C}} \Delta_k \|g_k\|. \quad (6.3.26)$$

Combining (6.3.25) and (6.3.26), we conclude that (6.3.19) holds with

$$\kappa_{\text{dep}} = \min[\kappa_{\text{ubs}}, 2\kappa_{\text{ubs}} \kappa_{\text{bck}} (1 - \kappa_{\text{ubs}})] < 1. \quad \square$$

Note that the Cauchy point may be different from the approximate Cauchy point even in the special case where the model is quadratic. This can be seen on the top left part of Figure 6.3.3.

### 6.3.4 The Final Condition on the Model Decrease

We see from Theorems 6.3.1 and 6.3.3 that the model reductions obtained at the Cauchy point or at the approximate Cauchy point have the same form. Because of this similarity and because both points play the same role in our theory, we abandon

the distinction between  $x_k^C$  and  $x_k^{AC}$ , the true and approximate versions of the Cauchy point. We retain only the name and notation of the former to cover both cases.

We also see that the model decrease at the Cauchy point depends on the value of  $\nu_k^C$ , at least for small trust-region radius  $\Delta_k$ . If we think of defining the trust region in the Euclidean norm, then (6.3.5) shows that  $\nu_k^C = 1$  for all  $k$ . If we consider other norms, AN.1 guarantees that

$$\nu_k^C \geq \frac{1}{\kappa_{\text{une}}} > 0$$

for all  $k$ .

In view of this bound, and since we may expect to decrease the model at least as much as at the Cauchy point, as given by (6.3.4) and (6.3.19), it is acceptable to require a model decrease of the following form.

**AA.1** For all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \Delta_k \right] \quad (6.3.27)$$

for some constant<sup>61</sup>  $\kappa_{\text{mdc}} \in (0, 1)$ .

This only implies that the total model decrease is at least a fraction of that obtained at the Cauchy point. This assumption has the following useful consequence.

**Theorem 6.3.4** Suppose that AM.3 and AA.1 hold and that  $\nabla_x f(x_k) \neq 0$ . Then  $m_k(x_k + s_k) < m_k(x_k)$  and  $s_k \neq 0$ .

**Proof.** By AM.3 we obtain that  $g_k \neq 0$ . Then AA.1 implies that  $m_k(x_k + s_k) < m_k(x_k)$ , as desired.  $\square$

Hence we see that the value of  $\rho_k$ , as defined by (6.1.4) (p. 116), is well defined, provided  $x_k$  is not a first-order critical point.

We may of course decide to decrease the model more than that implied by the bound given in AA.1, if the cost of the additional computation is acceptable. In particular, we may want to solve the subproblem

$$\min_{x \in \mathcal{B}_k} m_k(x) \quad (6.3.28)$$

exactly. We call an argument that gives the global solution of problem (6.3.28) a *model minimizer* and denote it by  $x_k^M$ . Figure 6.3.4 illustrates the contour lines of the model and the relative positions of the Cauchy point and the model minimizer in this case.

We note here that it is only necessary to find an approximation to the model minimizer to guarantee AA.1, as is shown by the following elementary result.

<sup>61</sup>“mdc” stands for “model decrease”. This assumption’s name (see page xviii) indicates that it is concerned with the algorithm.

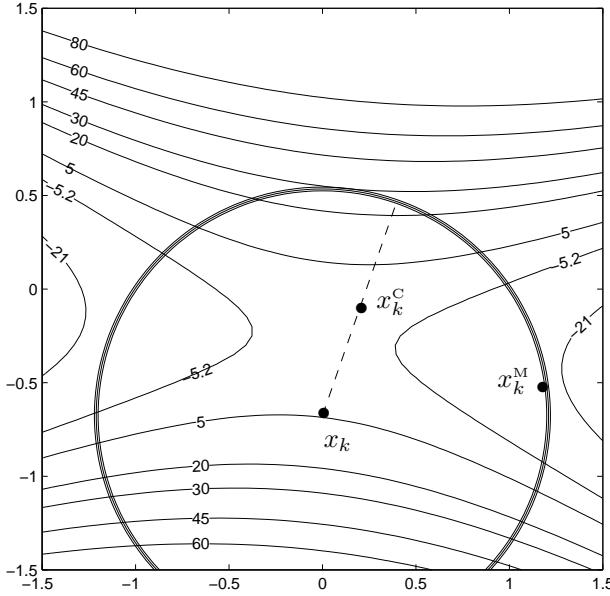


Figure 6.3.4: The Cauchy point  $x_k^C$  and the model minimizer  $x_k^M$  for a nonconvex model.

**Theorem 6.3.5** Suppose that, for all  $k$ , the step  $s_k$  ensures that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{amm}} [m_k(x_k) - m_k(x_k^M)],$$

where  $\kappa_{\text{amm}} \in (0, 1]$  is a constant.<sup>62</sup> Then AA.1 holds for suitably defined  $\kappa_{\text{mdc}}$ .

**Proof.** Clearly, we have that

$$m_k(x_k^M) \leq m_k(x_k^C)$$

and the choice  $s_k = x_k^M - x_k$  (using (6.3.4) and (6.3.19)) satisfies (6.3.27) with  $\kappa_{\text{mdc}} = \min[\frac{1}{2}, \kappa_{\text{dcp}}]/\kappa_{\text{une}}$ . Thus AA.1 follows, with  $\kappa_{\text{mdc}}$  replaced by  $\kappa_{\text{amm}} \min[\frac{1}{2}, \kappa_{\text{dcp}}]/\kappa_{\text{une}}$ .  $\square$

This result is important because it states that practical implementations based on the calculation of the model minimizer (see Section 7.2) fall within our framework and are thus covered by the convergence theory that we develop in the following sections.

We conclude our discussion of the model decrease by noting that we have chosen to define the Cauchy arc as the intersection of the steepest-descent direction and the trust region. However, we could equally well define it as the intersection of the steepest-descent direction with another ball  $B_k^C$ , centered at  $x_k$ , whose radius  $\Delta_k^C$  is such that the ratio  $\Delta_k/\Delta_k^C$  is bounded away from zero and from infinity. Of course, we have to

<sup>62</sup>“amm” stands for “approximate model minimizer”.

insist that AA.1 still holds, but this may be very easy to ensure, especially if we choose  $\mathcal{B}_k^c \subseteq \mathcal{B}_k$ . Furthermore, it is even possible to choose the norm defining the ball  $\mathcal{B}_k^c$  to be different from that defining  $\mathcal{B}_k$  without altering any of our results, provided all norms used remain uniformly equivalent.

## 6.4 Convergence to First-Order Critical Points

We now wish to prove that Algorithm BTR is globally convergent to first-order critical points. More precisely, we wish to prove that, under the assumptions stated in the previous sections, all limit points  $x_*$  of the sequence  $\{x_k\}$  generated by the algorithm are first-order critical for problem (6.1.1) (p. 115); that is, they satisfy

$$\nabla_x f(x_*) = 0$$

*irrespective of the position of the starting vector  $x_0$  and choice of the initial trust-region radius<sup>63</sup>  $\Delta_0$ .*

The first step of our analysis is to examine the size of the error between the true objective function and its model at a new iterate  $x_k + s_k \in \mathcal{B}_k$ .

**Theorem 6.4.1** Suppose that AF.1, AF.3, and AM.1–AM.4 hold. Then we have that, for all  $k$ ,

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq [\nu_k^s]^2 \max[\kappa_{\text{ufh}}, \kappa_{\text{umh}}] \Delta_k^2, \quad (6.4.1)$$

where  $x_k + s_k \in \mathcal{B}_k$  and

$$\nu_k^s = \frac{\|s_k\|}{\|s_k\|_k}. \quad (6.4.2)$$

Moreover, if AN.1 also holds, then

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_k^2, \quad (6.4.3)$$

where<sup>64</sup>

$$\kappa_{\text{ubh}} \stackrel{\text{def}}{=} \kappa_{\text{une}}^2 \max[\kappa_{\text{ufh}}, \kappa_{\text{umh}}]. \quad (6.4.4)$$

**Proof.** Using AF.1 and AM.1, we may apply the mean value theorem on the objective function and the model, and we obtain that

$$f(x_k + s_k) = f(x_k) + \langle s_k, \nabla_x f(x_k) \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle \quad (6.4.5)$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ , and, similarly, that

---

<sup>63</sup>First-order criticality, by its very nature, is not able to say anything about globality versus locality, a very desirable but equally elusive result in the absence of additional assumptions on the objective function.

<sup>64</sup>“ubh” stands for “upper bound on the Hessians”.

$$m_k(x_k + s_k) = m_k(x_k) + \langle s_k, g_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle \quad (6.4.6)$$

for some  $\zeta_k$  in the segment  $[x_k, x_k + s_k]$ . Subtracting (6.4.6) from (6.4.5) and taking absolute values yields that

$$\begin{aligned} |f(x_k + s_k) - m_k(x_k + s_k)| &= \frac{1}{2} |\langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle - \langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\ &\leq \frac{1}{2} |\langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle| + \frac{1}{2} |\langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\ &\leq \frac{1}{2} (\kappa_{\text{uuh}} + \kappa_{\text{umh}} - 1) \|s_k\|^2 \\ &= \frac{1}{2} (\kappa_{\text{uuh}} + \kappa_{\text{umh}} - 1) [\nu_k^s]^2 \|s_k\|^2 \\ &\leq \frac{1}{2} (\kappa_{\text{uuh}} + \kappa_{\text{umh}} - 1) [\nu_k^s]^2 \Delta_k^2, \end{aligned} \quad (6.4.7)$$

where we successively used AM.2, AM.3, AF.3, AM.4, the triangle and Cauchy–Schwarz inequalities, and the fact that  $x_k + s_k \in \mathcal{B}_k$  implies that  $\|s_k\|_k \leq \Delta_k$ . Thus (6.4.1) clearly holds, and (6.4.3) follows from AN.1 and the definitions (6.4.2) and (6.4.4).  $\square$

We therefore see that the error between the objective function and the model decreases quadratically with the trust-region radius. The smaller this radius becomes, the better the model approximates the objective, which intuitively guarantees that minimizing the model within a sufficiently small trust region will also decrease the objective function, as desired. We next show that this intuition is vindicated, in that an iteration must be successful if the current iterate is not first-order critical and the trust-region radius is small enough.

**Theorem 6.4.2** Suppose that AF.1, AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Suppose furthermore that  $g_k \neq 0$  and that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\| (1 - \eta_2)}{\kappa_{\text{ubh}}}. \quad (6.4.8)$$

Then iteration  $k$  is very successful and

$$\Delta_{k+1} \geq \Delta_k. \quad (6.4.9)$$

**Proof.** Observe first that the condition  $\eta_2 \in (0, 1)$  and the inequality  $0 < \kappa_{\text{mdc}} < 1$  imply that

$$\kappa_{\text{mdc}} (1 - \eta_2) < 1.$$

Thus conditions (6.4.8), (6.3.2), and (6.4.4) imply that

$$\Delta_k < \frac{\|g_k\|}{\beta_k}. \quad (6.4.10)$$

As a consequence, AA.1 immediately gives that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \Delta_k \right] = \kappa_{\text{mdc}} \|g_k\| \Delta_k. \quad (6.4.11)$$

On the other hand, we may apply Theorem 6.4.1 and deduce from (6.4.11), (6.4.3), (6.4.10), and (6.4.8) that

$$|\rho_k - 1| = \left| \frac{f(x_k + s_k) - m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \leq \frac{\kappa_{\text{ubh}}}{\kappa_{\text{mdc}} \|g_k\|} \Delta_k \leq 1 - \eta_2. \quad (6.4.12)$$

Therefore,  $\rho_k \geq \eta_2$  and the iteration is very successful. Furthermore, (6.1.5) (p. 116) ensures that (6.4.9) holds.  $\square$

As a consequence of this property, we may now prove that the radius cannot become too small as long as a first-order critical point is not approached. This property is crucial in that it ensures that progress of the algorithm is always possible (except at first-order critical points). It also indicates how small the trust-region radius has to be relative to  $\|g_k\|$  in order to guarantee success of the iteration.

**Theorem 6.4.3** Suppose that AF.1–AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Suppose furthermore that there exists a constant<sup>65</sup>  $\kappa_{\text{lbg}} > 0$  such that  $\|g_k\| \geq \kappa_{\text{lbg}}$  for all  $k$ . Then there is a constant<sup>66</sup>  $\kappa_{\text{lbd}} > 0$  such that

$$\Delta_k \geq \kappa_{\text{lbd}} \quad (6.4.13)$$

for all  $k$ .

**Proof.** Assume that iteration  $k$  is the first such that

$$\Delta_{k+1} \leq \frac{\gamma_1 \kappa_{\text{mdc}} \kappa_{\text{lbg}} (1 - \eta_2)}{\kappa_{\text{ubh}}}. \quad (6.4.14)$$

Then we have from (6.1.5) (p. 116) that  $\gamma_1 \Delta_k \leq \Delta_{k+1}$ , and hence

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \kappa_{\text{lbg}} (1 - \eta_2)}{\kappa_{\text{ubh}}}.$$

Our assumption on  $\|g_k\|$  then implies that (6.4.8) holds and thus that (6.4.9) is satisfied. But this contradicts the fact that iteration  $k$  is the first such that (6.4.14) holds, and our initial assumption is therefore impossible. This yields the desired conclusion with

$$\kappa_{\text{lbd}} = \frac{\gamma_1 \kappa_{\text{mdc}} \kappa_{\text{lbg}} (1 - \eta_2)}{\kappa_{\text{ubh}}}. \quad \square$$

---

<sup>65</sup>“lbg” stands for “lower bound on the gradient”.

<sup>66</sup>“lbd” stands for “lower bound on delta ( $\Delta$ )”.

The preceding two results are sufficient to establish the criticality of the unique limit point of the sequence of iterates when there are only finitely many successful iterations.

**Theorem 6.4.4** Suppose that AF.1, AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Suppose furthermore that there are only finitely many successful iterations. Then  $x_k = x_*$  for all sufficiently large  $k$  and  $x_*$  is first-order critical.

**Proof.** The mechanism of the algorithm ensures that  $x_* = x_{k_0+1} = x_{k_0+j}$  for all  $j > 0$ , where  $k_0$  is the index of the last successful iterate. Moreover, since all iterations are unsuccessful for sufficiently large  $k$ , (6.1.3) and (6.1.5) (p. 116) imply that the sequence  $\{\Delta_k\}$  converges to zero. If  $\|g_{k_0+1}\| > 0$ , Theorem 6.4.2 then implies that there must be a successful iteration of index larger than  $k_0$ , which is impossible. Hence  $\|g_{k_0+1}\| = 0$  and  $x_*$  is first-order critical.  $\square$

Having proved the desired convergence property for the case where  $\mathcal{S}$  is finite, we now restrict our attention, for the rest of Section 6.4, to the case where there are infinitely many successful iterations. In this case, we start by proving that at least one accumulation point of the sequence of iterates (when the sequence is infinite) must be first-order critical. The intuition behind this result is that, if  $\Delta_k$  is small enough, the model should approximate the function well, because of Theorem 6.4.1, but at the same time,  $\Delta_k$  cannot become too small if  $\|g_k\|$  is bounded away from zero, as we showed in Theorem 6.4.3. Hence convergence to a first-order critical point should occur.

**Theorem 6.4.5** Suppose that AF.1–AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Then one has that

$$\liminf_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0. \quad (6.4.15)$$

**Proof.** Assume, for the purpose of deriving a contradiction, that for all  $k$ ,

$$\|\nabla_x f(x_k)\| = \|g_k\| \geq \epsilon \quad (6.4.16)$$

for some  $\epsilon > 0$ , where we have used AM.3 to obtain the first equality. Now consider a successful iteration with index  $k$ . The fact that  $k \in \mathcal{S}$ , along with AA.1 and conditions (6.3.2), (6.4.16), and (6.4.13) then give that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 [m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}} \right]. \quad (6.4.17)$$

Summing now over all successful iterations from 0 to  $k$ , we obtain that

$$f(x_0) - f(x_{k+1}) = \sum_{\substack{j=0 \\ j \in \mathcal{S}}}^k [f(x_j) - f(x_{j+1})] \geq \sigma_k \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}} \right],$$

where  $\sigma_k$  is the number of successful iterations up to iteration  $k$ . But since there are infinitely many such iterations, we have that

$$\lim_{k \rightarrow \infty} \sigma_k = +\infty,$$

and the difference between  $f(x_0)$  and  $f(x_{k+1})$  is unbounded, which clearly contradicts the fact that, according to AF.2, the objective function is bounded below. Our assumption (6.4.16) must therefore be false, which yields (6.4.15).  $\square$

Theorem 6.4.5 is the usual first step in the convergence analysis of a trust-region algorithm. It implies that, if the sequence of iterates has limit points, then at least one of them satisfies the first-order necessary condition stated in Theorem 3.2.1 (p. 38). We now prove the stronger result that this is the case for *all* such limit points.

**Theorem 6.4.6** Suppose that AF.1–AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Then one has that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

**Proof.** Assume, for the purpose of establishing a contradiction, that there is a subsequence of successful iterates, indexed by  $\{t_i\} \subseteq \mathcal{S}$ , such that

$$\|\nabla_x f(x_{t_i})\| = \|g_{t_i}\| \geq 2\epsilon > 0 \quad (6.4.18)$$

for some  $\epsilon > 0$  and for all  $i$ , where the first equality results from AM.3. Theorem 6.4.5 then ensures the existence for each  $t_i$  of a first successful iteration  $\ell(t_i) > t_i$  such that  $\|g_{\ell(t_i)}\| < \epsilon$ . Denoting  $\ell_i \stackrel{\text{def}}{=} \ell(t_i)$ , we thus obtain that there is another subsequence of  $\mathcal{S}$  indexed by  $\{\ell_i\}$  such that

$$\|g_k\| \geq \epsilon \text{ for } t_i \leq k < \ell_i \text{ and } \|g_{\ell_i}\| < \epsilon. \quad (6.4.19)$$

We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\text{def}}{=} \{k \in \mathcal{S} \mid t_i \leq k < \ell_i\},$$

where  $t_i$  and  $\ell_i$  belong to the two subsequences defined above. An illustration of the definition of the subsequences  $\{t_i\}$ ,  $\{\ell_i\}$ , and  $\mathcal{K}$  is presented in Figure 6.4.1, where it is assumed for simplicity that all iterations are successful. In this figure, we have marked position  $j$  in each of the subsequences represented in the abscissa when  $j$  belongs to that subsequence. Note in this example that  $\ell_0 = \ell_1 = \ell_2 = \ell_3 = \ell_4 =$

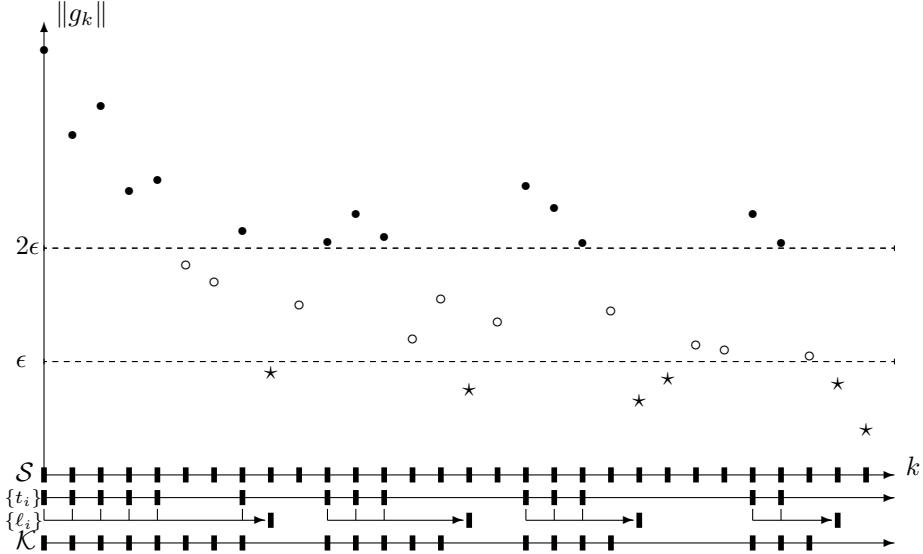


Figure 6.4.1: The subsequences of the proof of Theorem 6.4.6.

$\ell_5 = 8$ , which we indicated by arrows from  $t_0 = 0$ ,  $t_1 = 1$ ,  $t_2 = 2$ ,  $t_3 = 3$ ,  $t_4 = 4$ , and  $t_5 = 7$  to  $k = 9$ , and so on.

Using AA.1, the fact that  $\mathcal{K} \subseteq \mathcal{S}$ , and (6.4.19), we deduce that for  $k \in \mathcal{K}$ ,

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_k \right], \quad (6.4.20)$$

where we have also used (6.3.2) to obtain the second inequality. But the sequence  $\{f(x_k)\}$  is monotonically decreasing and bounded below because of AF.2. Hence it is convergent, and the left-hand side of (6.4.20) must tend to zero when  $k$  tends to infinity. This gives that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \Delta_k = 0.$$

As a consequence, the second term dominates in the minimum of (6.4.20) and we obtain that, for  $k \in \mathcal{K}$  sufficiently large,

$$\Delta_k \leq \frac{1}{\kappa_{\text{mdc}} \epsilon \eta_1} [f(x_k) - f(x_{k+1})]. \quad (6.4.21)$$

We then deduce from this bound that, for  $i$  sufficiently large,

$$\|x_{t_i} - x_{\ell_i}\| \leq \sum_{j=t_i}^{\ell_i-1} \|x_j - x_{j+1}\| \leq \sum_{j=t_i}^{\ell_i-1} \nu_j^s \Delta_j \leq \frac{\kappa_{\text{une}}}{\kappa_{\text{mdc}} \epsilon \eta_1} [f(x_{t_i}) - f(x_{\ell_i})], \quad (6.4.22)$$

where we also used AN.1 and (6.4.2) to derive the last inequality. Because of AF.2 and the monotonicity of the sequence  $\{f(x_k)\}$  again, we see that the right-hand side of this inequality must converge to zero, and we therefore obtain that  $\|x_{t_i} - x_{\ell_i}\|$  tends to zero as  $i$  tends to infinity. By continuity of the gradient AF.1, we thus

deduce that  $\|\nabla_x f(x_{t_i}) - \nabla_x f(x_{\ell_i})\|$  and, by AM.3,  $\|g_{t_i} - g_{\ell_i}\|$  also tend to zero. However, this is impossible because of the definitions of  $\{t_i\}$  and  $\{\ell_i\}$ , which imply that  $\|g_{t_i} - g_{\ell_i}\| \geq \epsilon$ . Hence, no subsequence satisfying (6.4.18) can exist, and the theorem is proved.  $\square$

## Notes and References for Section 6.4

The first proofs of global convergence for trust-region methods in unconstrained optimization are due to Powell (1970c, 1970d, 1975). In particular, these papers introduce what we now call the Cauchy point (in recognition of the important work of Cauchy, 1847, on the steepest-descent method for the solution of systems of equations), the first formulation of AA.1, and Theorem 6.3.1. Powell also proved the important Theorem 6.4.5. However, the class of methods considered by Powell is slightly different from Algorithm BTR, notably because it chooses  $\eta_1 = 0$ . This implies that the trial step is accepted as soon as it produces a reduction in the objective function, as opposed to Algorithm BTR, where a “sufficient reduction” is required. The requirement that  $\eta_1$  should be strictly positive was first made by Thomas (1975), who proved that not only one but all limit points of the sequence of iterates are first-order critical (that is Theorem 6.4.6) when this additional restriction is imposed. In practice, the difference in behaviour between algorithms using  $\eta_1 = 0$  and  $\eta_1 > 0$  is negligible. Recently Yuan (1998b) showed that an algorithm using  $\eta_1 = 0$  could produce a sequence  $\{x_k\}$  for which some limit points are not first-order critical (although one of them must be because of Theorem 6.4.5). Another difference between the class of algorithms considered in the early papers by Powell and Algorithm BTR is that Powell required that the step  $s_k$  be the Newton step  $-H_k^{-1}g_k$  for positive definite  $H_k$ , whenever  $\|s_k\| < \Delta_k$  (assuming  $\nu_k^S = 1$ ). Fortunately, this assumption is not necessary for Algorithm BTR, and this allows the use of an approximate solution of the system  $H_k s_k = -g_k$  in the calculation of  $s_k$ . This is very useful when the dimension of the problem is large, as it allows the use of iterative methods such as conjugate gradients (see Section 5.1) to obtain an approximate solution of this system.

## 6.5 Second-Order Convex Models

The best we can hope for if we only require that our model coincides with the objective function up to first order is that our Algorithm BTR converges to a first-order critical point. We now consider exploiting second-order information to a larger extent, so as to deduce convergence to second-order critical points, that is, points  $x_*$  at which the second-order necessary optimality conditions

$$\nabla_x f(x_*) = 0 \text{ and } \nabla_{xx} f(x_*) \text{ is positive semidefinite}$$

(see Theorem 3.2.2, p. 38) are satisfied. In this section, we investigate the case where the models are asymptotically convex, while the more general nonconvex situation will be considered in Section 6.6.

Our first step is to investigate under what condition we may ensure not only that the sequence  $\{g_k\}$  converges to zero but also that the sequence of iterates  $\{x_k\}$  itself converges to a (single) limit point. As we now show, this depends on the second-order part of the model. We first prove a useful technical result bounding the size of the step as a function of the norm of the model gradient.

**Lemma 6.5.1** Suppose that AM.1 holds, and that

$$\lambda_{\min}[\nabla_{xx}m_k(x)] \geq \epsilon \quad (6.5.1)$$

for all  $x \in [x_k, x_k + s_k]$  and for some  $\epsilon > 0$ . Then

$$\|s_k\| \leq \frac{2}{\epsilon} \|g_k\|. \quad (6.5.2)$$

**Proof.** Consider the model decrease obtained at  $x_k + s_k$ , which is given by

$$m_k(x_k) - m_k(x_k + s_k) = -\langle g_k, s_k \rangle - \frac{1}{2}\langle s_k, \nabla_{xx}m_k(\xi_k)s_k \rangle$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ . Assume first that  $m_k(x_k + s_k) = m_k(x_k)$ . Then Theorem 6.3.4 and AM.3 imply that  $g_k = \nabla_x f(x_k) = 0$ . Thus we have that  $\langle s_k, \nabla_{xx}m_k(\xi_k)s_k \rangle = 0$  and we deduce from (6.5.1) that  $s_k = 0$ , which obviously yields (6.5.2). If, on the other hand,  $m_k(x_k + s_k) < m_k(x_k)$ , we have that  $s_k \neq 0$ . We then define

$$\phi(t) \stackrel{\text{def}}{=} m_k(x_k) - m_k(x_k + ts_k) = -t\langle g_k, s_k \rangle - \frac{1}{2}t^2\langle s_k, \nabla_{xx}m_k(\xi_k)s_k \rangle$$

for  $t > 0$ . But (6.5.1) and the fact that  $s_k \neq 0$  ensure that  $\langle s_k, \nabla_{xx}m_k(\xi_k)s_k \rangle > 0$ , and hence  $\phi$  is a concave quadratic function, as shown in Figure 6.5.1. Moreover,  $\phi(0) = 0$  and  $\phi(1) > 0$  by construction. As a consequence,

$$t_* = \arg \max_t \phi(t) \geq \frac{1}{2},$$

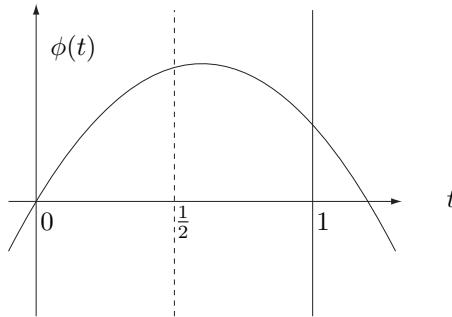


Figure 6.5.1: An example for the parabola  $\phi(t)$ .

where we used the symmetry of a quadratic with respect to its maximum to derive the last inequality (see Figure 6.5.1). But an easy computation shows that

$$t_* = \frac{|\langle g_k, s_k \rangle|}{\langle s_k, \nabla_{xx} m_k(\xi_k) s_k \rangle} \leq \frac{\|g_k\|}{\epsilon \|s_k\|},$$

where we have used the Cauchy–Schwarz inequality and (6.5.1) to derive the last part of this bound. Therefore, we deduce that (6.5.2) holds.  $\square$

We now examine the behaviour of the sequence generated by Algorithm BTR when the models are asymptotically convex along some subsequence  $\{k_i\}$ .

**Theorem 6.5.2** Suppose that AF.1–AF.3, AM.1–AM.4, and AA.1 hold. Suppose furthermore that  $\{x_{k_i}\}$  is a subsequence of iterates converging to the first-order critical point  $x_*$  and that there is a constant<sup>67</sup>  $\kappa_{\text{smh}} > 0$  such that

$$\min_{x \in \mathcal{B}_k} \lambda_{\min}[\nabla_{xx} m_k(x)] \geq \kappa_{\text{smh}} \quad (6.5.3)$$

whenever  $x_k$  is sufficiently close to  $x_*$ . Suppose finally that  $\nabla_{xx} f(x_*)$  is nonsingular. Then the complete sequence of iterates  $\{x_k\}$  converges to  $x_*$ .

**Proof.** The criticality of  $x_*$  is ensured by Theorem 6.4.6. If there are only finitely many successful iterations, then the desired conclusion immediately follows from Theorem 6.4.4. Otherwise, we may assume, without loss of generality, that the subsequence  $\{k_i\}$  consists only of successful iterations, which yields that

$$x_{k_i+1} = x_{k_i} + s_{k_i} \quad (6.5.4)$$

for all  $i$ . In order to prove the convergence of the complete sequence, we choose a  $\delta > 0$  whose value is small enough to ensure that (6.5.3) holds for all  $k$  such that  $\|x_k - x_*\| \leq \delta$  and that

$$\|\nabla_{xx} f(x) - \nabla_{xx} f(x_*)\| \leq \frac{1}{4} \min [\sigma, 1, \kappa_{\text{smh}}] \stackrel{\text{def}}{=} \delta_0 \quad (6.5.5)$$

for all  $x$  such that  $\|x - x_*\| \leq \delta$ , where  $\sigma > 0$  is the smallest singular value of  $\nabla_{xx} f(x_*)$ . This last bound is possible because of the nonsingularity of  $\nabla_{xx} f(x_*)$  and the continuity of the Hessian of the objective function (AF.1). We may also choose  $i_1$  large enough to ensure that

$$\|x_{k_i} - x_*\| \leq \frac{\kappa_{\text{smh}} \delta}{2\delta_0 + \kappa_{\text{smh}}} \stackrel{\text{def}}{=} \delta_1 \quad (6.5.6)$$

for all  $i \geq i_1$  and also that

$$\|g_k\| \leq \delta_0 \delta_1 < \delta \quad (6.5.7)$$

---

<sup>67</sup>“smh” stands for “smallest eigenvalue of the model’s Hessian”.

for all  $k \geq k_{i_1}$ . Inequality (6.5.6) is possible because we assumed that  $\{x_{k_i}\}$  converges to  $x_*$ , and (6.5.7) because of Theorem 6.4.6 and the inequalities

$$\delta_0 < 1 \text{ and } \delta_1 < \delta,$$

themselves resulting from the definitions of  $\delta_0$  and  $\delta_1$  in (6.5.5) and (6.5.6), respectively.

We may now apply Lemma 6.5.1 at iteration  $k_i$  with  $\epsilon = \kappa_{\text{smh}}$  and deduce that

$$\|s_{k_i}\| \leq \frac{2}{\kappa_{\text{smh}}} \|g_{k_i}\|. \quad (6.5.8)$$

Hence, combining (6.5.4), (6.5.6), (6.5.7), and (6.5.8), we obtain that

$$\|x_{k_i+1} - x_*\| \leq \|x_{k_i} - x_*\| + \|s_{k_i}\| \leq \left(1 + \frac{2\delta_0}{\kappa_{\text{smh}}}\right) \delta_1 = \delta. \quad (6.5.9)$$

Assume now that

$$\|x_{k_i+1} - x_*\| > \delta_1 \quad (6.5.10)$$

and observe that, because of AM.3,

$$g_{k_i+1} = \nabla_x f(x_{k_i+1}) = \nabla_x f(x_*) + G_*(x_{k_i+1} - x_*),$$

where

$$G_* = \int_0^1 \nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) dt. \quad (6.5.11)$$

Hence, using the triangle inequality, the definition of  $\sigma$ , (6.5.9), the fact that  $\nabla_x f(x_*) = 0$ , and (6.5.10),

$$\begin{aligned} \|g_{k_i+1}\| &= \|\nabla_{xx} f(x_*)(x_{k_i+1} - x_*) + (G_* - \nabla_{xx} f(x_*)(x_{k_i+1} - x_*))\| \\ &\geq \|\nabla_{xx} f(x_*)(x_{k_i+1} - x_*)\| - \|G_* - \nabla_{xx} f(x_*)\| \|(x_{k_i+1} - x_*)\| \\ &> \sigma\delta_1 - \|G_* - \nabla_{xx} f(x_*)\|\delta. \end{aligned} \quad (6.5.12)$$

But (6.5.11) and (6.5.5) then give that

$$\begin{aligned} \|G_* - \nabla_{xx} f(x_*)\| &= \left\| \int_0^1 [\nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) - \nabla_{xx} f(x_*)] dt \right\| \\ &\leq \max_{t \in [0,1]} \|\nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) - \nabla_{xx} f(x_*)\| \\ &\leq \delta_0. \end{aligned} \quad (6.5.13)$$

Therefore, combining (6.5.12), (6.5.13), the definition of  $\delta_1$  in (6.5.6), and the definition of  $\delta_0$  in (6.5.5), we deduce that

$$\|g_{k_i+1}\| > \sigma\delta_1 - \delta_0 \geq 4\delta_0\delta_1 - \delta_0\delta_1 \frac{2\delta_0 + \kappa_{\text{smh}}}{\kappa_{\text{smh}}} = \frac{\delta_0\delta_1(3\kappa_{\text{smh}} - 2\delta_0)}{\kappa_{\text{smh}}} > \delta_0\delta_1.$$

This is impossible because of (6.5.7), and therefore

$$\|x_{k_i+1} - x_*\| \leq \delta_1 < \delta.$$

All the conditions that are satisfied at  $x_{k_i}$  thus remain satisfied at  $x_{k_i+1}$ , and the argument can be applied recursively to show that, for all  $j \geq 1$ ,

$$\|x_{k_i+j} - x_*\| \leq \delta_1 < \delta.$$

Since  $\delta$  is arbitrarily small, this proves the convergence of the complete sequence  $\{x_k\}$  to  $x_*$ .  $\square$

This theorem has both a positive and a negative side. On the positive side, we see that if there is a subsequence converging to an isolated minimizer and if the model reflects this situation correctly, then the whole sequence converges to that minimizer. On the negative side, we observe that it is enough that the model is uniformly convex along a subsequence converging to an isolated first-order critical point (possibly a saddle point), and thus convergence of the complete sequence occurs, even if the convexity of the model doesn't reflect the true geometry of the objective function. This undesirable situation is illustrated in Figure 6.5.2, where the model, whose contour lines are shown inside the trust region, is convex, while  $x_k$  is close to a saddle point, as indicated by the contour lines of the objective function (outside the trust region).

Our next result investigates the positive side of Theorem 6.5.2 a little further. In accordance with our discussion of the previous paragraph, we will assume not only that the limit point is an isolated minimizer of the objective function but also that the model and objective function coincide up to second order, whenever the iterates approach a first-order critical point. More precisely, we phrase this assumption as follows.

**AM.5** We assume that

$$\lim_{k \rightarrow \infty} \|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| = 0 \text{ whenever } \lim_{k \rightarrow \infty} \|g_k\| = 0. \quad (6.5.14)$$

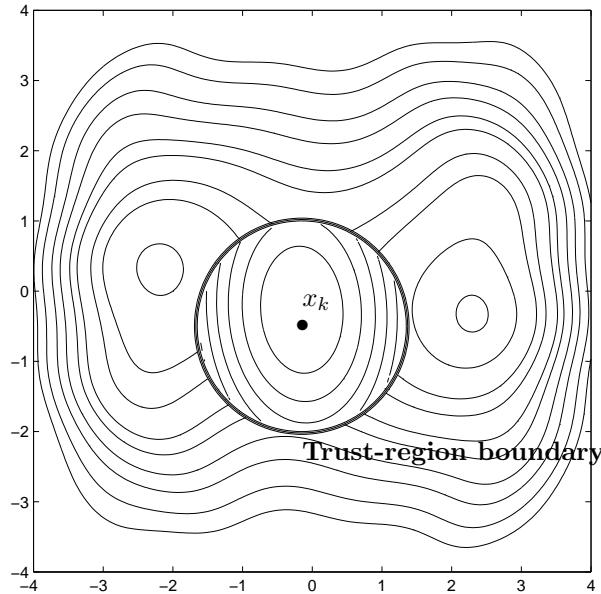


Figure 6.5.2: An example where a convex model for a nonconvex function may force convergence to a saddle point.

This assumption aims to avoid situations like that of Figure 6.5.2: Figure 6.5.3 shows a (nonconvex) model obeying AM.5. Note that Theorem 6.4.6 ensures that the second part of (6.5.14) always holds. Also note that AF.3, AM.5, and AF.1 together imply AM.4 if the iterate remains in a bounded domain. Thus, this last additional assumption allows us to ignore AM.4 as long as we assume AM.5.

Given this assumption on the quality of the model, we are now in position to analyse its implications in the context of our theory. We start with two simple technical lemmas, which will be used both here and in later sections.

**Lemma 6.5.3** Suppose that AF.1–AF.3, AM.1–AM.3, and AM.5 hold. Suppose also that there exist a subsequence  $\{k_i\}$  and a constant<sup>68</sup>  $\kappa_{\text{mqd}} > 0$  such that

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{\text{mqd}} \|s_{k_i}\|^2 > 0 \quad (6.5.15)$$

for all  $i$  sufficiently large. Finally, suppose that

$$\lim_{i \rightarrow \infty} \|s_{k_i}\| = 0.$$

Then

$$\rho_{k_i} \geq \eta_2$$

for  $i$  sufficiently large.

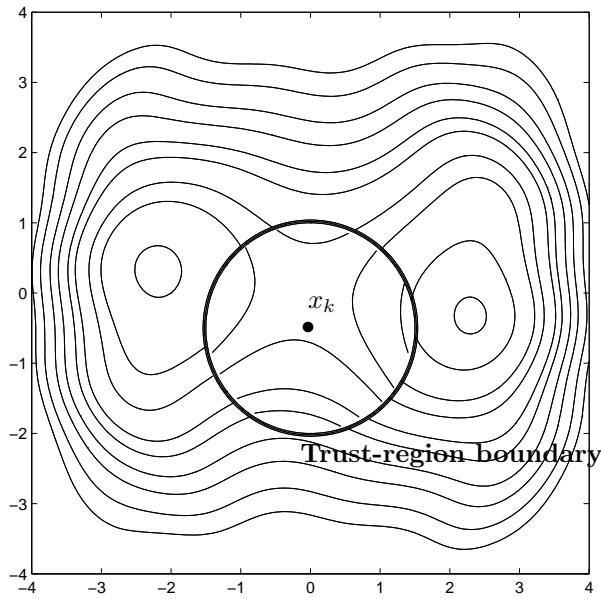


Figure 6.5.3: An example where a nonconvex model for a nonconvex function allows the iterates to leave a saddle point.

<sup>68</sup>“mqd” stands for “model quadratic decrease”.

**Proof.** The mean value theorem implies that, for  $k$  sufficiently large and for some  $\xi_{k_i}$  and  $\zeta_{k_i}$  in the segment  $[x_{k_i}, x_{k_i} + s_{k_i}]$ ,

$$\begin{aligned} |\rho_{k_i} - 1| &= \left| \frac{f(x_{k_i} + s_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i})}{m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i})} \right| \\ &\leq \frac{1}{\kappa_{\text{mqd}} \|s_{k_i}\|^2} |\langle s_{k_i}, \nabla_{xx} f(\xi_{k_i}) s_{k_i} \rangle - \langle s_{k_i}, \nabla_{xx} m_{k_i}(\zeta_{k_i}) s_{k_i} \rangle| \\ &= \frac{1}{\kappa_{\text{mqd}} \|s_{k_i}\|^2} |\langle s_{k_i}, [\nabla_{xx} f(\xi_{k_i}) - \nabla_{xx} m_{k_i}(\zeta_{k_i})] s_{k_i} \rangle| \\ &\leq \frac{1}{\kappa_{\text{mqd}} \|s_{k_i}\|^2} \|s_{k_i}\|^2 \|\nabla_{xx} f(\xi_{k_i}) - \nabla_{xx} m_{k_i}(\zeta_{k_i})\| \\ &\leq \frac{1}{\kappa_{\text{mqd}}} \left[ \|\nabla_{xx} f(\xi_{k_i}) - \nabla_{xx} f(x_{k_i})\| + \|\nabla_{xx} f(x_{k_i}) - \nabla_{xx} m_{k_i}(x_{k_i})\| \right. \\ &\quad \left. + \|\nabla_{xx} m_{k_i}(x_{k_i}) - \nabla_{xx} m_{k_i}(\zeta_{k_i})\| \right], \end{aligned}$$

where we also used (6.5.15), the triangle inequality, and the Cauchy–Schwarz inequality. But the first and third terms of the last right-hand side tend to zero when  $i$  goes to infinity because of AF.1, AM.1, our assumption that  $\|s_{k_i}\|$  tends to zero, and the bounds

$$\|\xi_{k_i} - x_{k_i}\| \leq \|s_{k_i}\| \text{ and } \|\zeta_{k_i} - x_{k_i}\| \leq \|s_{k_i}\|.$$

The second is also arbitrarily small because of AM.5 and Theorem 6.4.6. As a consequence,  $\rho_{k_i}$  tends to 1 when  $i$  tends to infinity, and is thus larger than  $\eta_2$  for  $i$  sufficiently large.  $\square$

This lemma has the following easy consequence.

**Lemma 6.5.4** Suppose that AF.1–AF.3, AM.1–AM.3, and AM.5 hold. Suppose also that there exists a constant  $\kappa_{\text{mqd}} > 0$  such that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mqd}} \|s_k\|^2 > 0 \quad (6.5.16)$$

for all  $k$  sufficiently large. Finally, suppose that

$$\lim_{k \rightarrow \infty} \|s_k\| = 0.$$

Then all iterations are eventually very successful and  $\Delta_k$  is bounded away from zero.

**Proof.** We first apply Lemma 6.5.3 to the complete sequence and obtain that all iterations are eventually very successful. Since the mechanism of Algorithm BTR implies that the trust-region radius is never decreased following very successful iterations, we then deduce that  $\Delta_k$  is bounded away from zero.  $\square$

Using this technical result, we may now prove that, if one of the limit points is an isolated minimizer, then the complete sequence converges to that minimizer and the trust-region radius  $\Delta_k$  eventually becomes irrelevant in the calculation of the step.

**Theorem 6.5.5** Suppose that AF.1–AF.3, AN.1, AM.1–AM.5, and AA.1 hold and that  $\{x_{k_i}\}$  is a subsequence of the iterates generated by Algorithm BTR converging to the first-order critical point  $x_*$ . Suppose furthermore that  $s_k \neq 0$  for all  $k$  sufficiently large. Finally, suppose that  $\nabla_{xx}f(x_*)$  is positive definite. Then the complete sequence of iterates  $\{x_k\}$  converges to  $x_*$ , all iterations are eventually very successful, and the trust-region radius  $\Delta_k$  is bounded away from zero.

**Proof.** Observe first that Theorem 6.4.6, AM.5, and the positive definiteness of  $\nabla_{xx}f(x_*)$  imply that  $\nabla_{xx}m_{k_i}(x)$  is positive definite for  $i$  large enough and for all  $x \in \mathcal{B}_{k_i}$ , where the subsequence  $\{k_i\}$  is chosen as the index set of any subsequence of successful iterates converging to  $x_*$ . Hence, there must exist a constant  $\kappa_{\text{smh}} > 0$  such that (6.5.3) holds for any such subsequence. We may thus apply Theorem 6.5.2 and deduce that the complete sequence  $\{x_k\}$  is converging to  $x_*$ . We also obtain from Lemma 6.5.1 that

$$\|g_k\| \geq \frac{\kappa_{\text{smh}}}{2} \|s_k\| > 0 \quad (6.5.17)$$

for  $k$  sufficiently large. Hence AA.1 and the inequalities  $\|s_k\| = \nu_k^S \|s_k\|_k \leq \nu_k^S \Delta_k$ ,  $\beta_k \leq \kappa_{\text{umh}}$ , and  $\nu_k^S \leq \kappa_{\text{une}}$  (by AN.1) yield that, for  $k$  sufficiently large,

$$\begin{aligned} m_k(x_k) - m_k(x_k + s_k) &\geq \frac{1}{2} \kappa_{\text{smh}} \kappa_{\text{mdc}} \|s_k\| \min \left[ \frac{\kappa_{\text{smh}} \|s_k\|}{2\beta_k}, \Delta_k \right] \\ &\geq \frac{1}{2} \kappa_{\text{smh}} \kappa_{\text{mdc}} \|s_k\|^2 \min \left[ \frac{\kappa_{\text{smh}}}{2\beta_k}, \frac{1}{\kappa_{\text{une}}} \right] \\ &\geq \delta \|s_k\|^2, \end{aligned} \quad (6.5.18)$$

where

$$\delta = \frac{1}{2} \kappa_{\text{smh}} \kappa_{\text{mdc}} \min \left[ \frac{\kappa_{\text{smh}}}{2\kappa_{\text{umh}}}, \frac{1}{\kappa_{\text{une}}} \right].$$

Furthermore, since  $g_k$  tends to zero as stated by Theorem 6.4.6, we also deduce from (6.5.17) that

$$\lim_{k \rightarrow \infty} \|s_k\| = 0.$$

This limit, our assumption on  $s_k$ , and (6.5.18) then allow us to apply Lemma 6.5.4 with  $\kappa_{\text{mqd}} = \delta$  and obtain the desired conclusions.  $\square$

A very important conclusion of Theorem 6.5.5 and the uniform equivalence between the Euclidean and  $\|\cdot\|_k$  norms is that the length of the step is not asymptotically constrained by the trust-region constraint

$$\frac{\|s_k\|}{\kappa_{\text{une}}} \leq \|s_k\|_k \leq \Delta_k,$$

since the step must tend to zero if  $\{x_k\}$  converges to  $x_*$  while  $\Delta_k$  is bounded away from zero. This means that *the rate of convergence of Algorithm BTR is entirely determined by the precise method used to compute the step when the trust-region constraint is inactive*. For instance, the rate of convergence of a trust-region variant of Newton's method (where the model is chosen to be the first three terms of a Taylor expansion of the objective function and is minimized exactly if  $\|s_k\|_k < \Delta_k$ ) is quadratically convergent. Although the analysis of the local rate of convergence of optimization algorithms is an important and useful subject, it is not central to the theme of this book, and we will pursue it no further.

## Notes and References for Section 6.5

Theorem 6.5.2 seemingly appeared for the first time in Conn et al. (1993), although in the more general context of problems with convex constraints. Theorem 6.5.5 was pointed out by Sorensen (1982a) and Moré and Sorensen (1983).

## 6.6 The Eigenpoint and Second-Order Nonconvex Models

We now wish to investigate whether it is possible to prove convergence of the sequence of iterates to second-order critical points of  $f$ , that is, to points where the second-order necessary conditions hold, without assuming positive definiteness of the Hessian of the objective function at the limit points. Note that second-order critical points may be weak minimizers, but need not be, as is illustrated by Figure 6.6.1.

The left part of this figure shows the contour lines of the function  $\frac{1}{2} \max[0, (x_1 - 1)^3, -(x_1 + 1)^3] + x_2^2$ , and the right part, those of  $x_1^3(x_1 - 1) + x_2^4$ , a function that has a

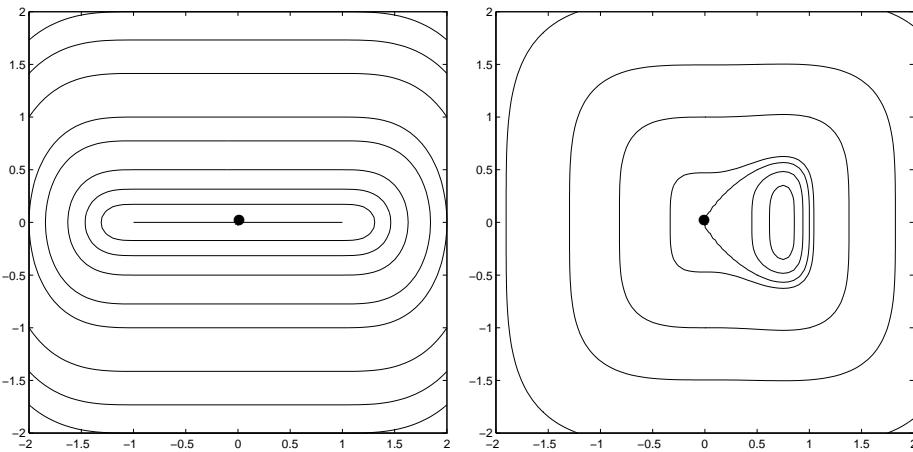


Figure 6.6.1: Second-order critical points that are not second-order sufficient.

single minimum around  $(0.7, 0)$ . Both functions are twice-continuously differentiable. In both cases, the origin (indicated with a black dot) is a second-order critical point but is only a weak minimizer in the first, while it is a multidimensional inflexion point in the second.

Intuitively, convergence to second-order critical points is only possible if the algorithm somehow avoids converging to saddle points or maximizers. One way to avoid converging to such points is to take advantage of directions of negative curvature when they exist and when the steps are becoming small. This obviously requires that the algorithm “sees” negative curvature when it is significant and incorporates it into the model (which is ensured by AM.5), and also that negative curvature in the model is then exploited.

### 6.6.1 Exploitation of Negative Curvature by Minimization

As in Section 6.3, we start by considering a quadratic model and analyse what model reduction can be achieved if the step uses negative curvature, when present. Since the model is quadratic, it is straightforward to calculate its exact minimizer within the trust region in any given direction, a feature that we exploit in what follows.

When  $H_k = \nabla_{xx} m_k(x_k)$  has a (strictly) negative eigenvalue  $\tau_k$ , which we assume to be the case in this section and the following one, we may determine (admittedly, usually at some cost; see Section 7.5.5) a direction  $u_k$  such that

$$\langle u_k, g_k \rangle \leq 0, \quad \|u_k\|_k = \Delta_k, \quad \text{and} \quad \langle u_k, H_k u_k \rangle \leq \kappa_{\text{snc}} \tau_k [\nu_k^{\text{E}}]^2 \Delta_k^2 \quad (6.6.1)$$

for some constant<sup>69</sup>  $\kappa_{\text{snc}} \in (0, 1]$  and where

$$\nu_k^{\text{E}} \stackrel{\text{def}}{=} \frac{\|u_k\|}{\|u_k\|_k}. \quad (6.6.2)$$

This means that  $u_k$  has a significant component along the eigenvectors of  $H_k$  corresponding to negative eigenvalues. It may practically be chosen as an approximate eigenvector of  $H_k$  corresponding to the eigenvalue  $\tau_k$ , whose sign and scale are chosen to ensure the first two parts of (6.6.1). We then wish to minimize our quadratic model within the intersection of the multiples of  $u_k$  and the trust region  $\mathcal{B}_k$ , much in the same vein as for the determination of the Cauchy point. Formally, we calculate

$$x_k^{\text{E}} = x_k + t_k^{\text{E}} u_k \in \mathcal{B}_k$$

such that

$$m_k(x_k^{\text{E}}) = m_k(x_k + t_k^{\text{E}} u_k) = \min_{t \in (0, 1]} m_k(x_k + tu_k). \quad (6.6.3)$$

This minimum is obviously achieved for  $t_k^{\text{E}} = 1$ . The point  $x_k^{\text{E}}$ , which we call the *eigenpoint*, plays a role similar to that of the Cauchy point, in that we now require the overall model decrease at  $x_k + s_k$  to satisfy

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{nmd}} (m_k(x_k) - \min[m_k(x_k^{\text{C}}), m_k(x_k^{\text{E}})]), \quad (6.6.4)$$

---

<sup>69</sup>“snc” stands for “sufficient negative curvature”.

where  $\kappa_{\text{nmd}} \in (0, 1]$  is a constant.<sup>70</sup> This requirement can easily be satisfied by choosing the appropriate one of

$$x_k + s_k = x_k^C \text{ or } x_k + s_k = x_k^E$$

or any other point in  $\mathcal{B}_k$  which reduces the model even further.

An example of the determination of the Cauchy and eigenpoints is illustrated in Figure 6.6.2 for an indefinite model. In the case shown, the model reduction at the eigenpoint is larger than that at the Cauchy point, and (6.6.4) will prevent the algorithm from converging to the saddle point.

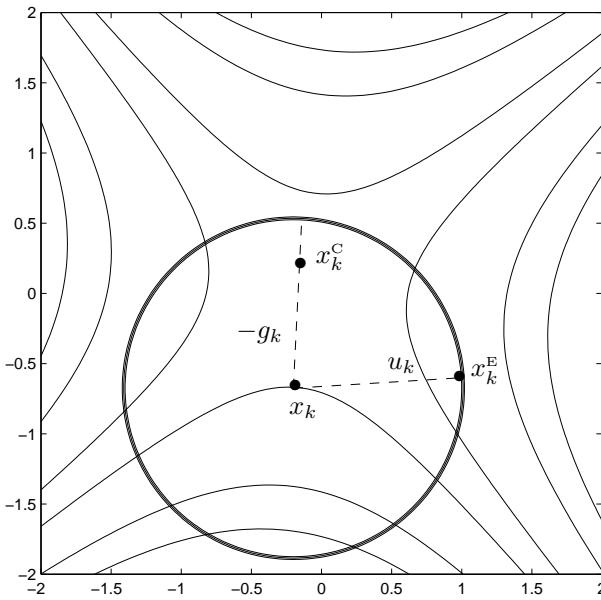


Figure 6.6.2: The Cauchy and eigenpoints for an indefinite quadratic model.

We may now analyse the corresponding model decrease at the eigenpoint.

**Theorem 6.6.1** Suppose that  $m_k$  is given by (6.1.7) and that  $H_k$  has a negative eigenvalue  $\tau_k$ . Then we have that

$$m_k(x_k) - m_k(x_k^E) \geq -\frac{1}{2}\kappa_{\text{snc}}\tau_k[\nu_k^E]^2\Delta_k^2, \quad (6.6.5)$$

where  $\nu_k^E$  is defined by (6.6.2).

**Proof.** We observe that the assumptions of the theorem and (6.6.1) imply that

$$\langle \nabla_x m_k(x_k + tu_k), u_k \rangle = \langle g_k, u_k \rangle + t\langle u_k, H_k u_k \rangle < 0$$

<sup>70</sup>“nmd” stands for “negative curvature model decrease”.

for all  $t \in (0, 1]$ . Hence the minimum of (6.6.3) must lie on the boundary of the trust region. As a consequence, we obtain that

$$t_k^E = 1.$$

Using this equality and (6.6.1) in the model's expression then gives that

$$\begin{aligned} m_k(x_k) - m_k(x_k^E) &= -t_k^E \langle u_k, g_k \rangle - \frac{1}{2} [t_k^E]^2 \langle u_k, H_k u_k \rangle \\ &\geq -\frac{1}{2} \langle u_k, H_k u_k \rangle \\ &\geq -\frac{1}{2} \kappa_{\text{snc}} \tau_k [\nu_k^E]^2 \Delta_k^2, \end{aligned}$$

as required.  $\square$

Combining (6.6.4), (6.3.4), and (6.6.5), we obtain that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{nmd}} \max \left\{ \frac{1}{2} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \nu_k^C \Delta_k \right], -\frac{1}{2} \kappa_{\text{snc}} \tau_k [\nu_k^E]^2 \Delta_k^2 \right\} \quad (6.6.6)$$

when we use exact minimization along an (approximate) eigenvector to determine the eigenpoint.

### 6.6.2 Exploitation of Negative Curvature by a Linesearch

We now return to the more general case where we do not require the model to be quadratic. This creates the difficulty that usually the eigenpoint can no longer be calculated exactly. Furthermore the curvature of the model is, in general, no longer constant along the step, which means that the minimum of the model along this direction (if we could compute it, as in (6.6.3)) might well occur in the interior of the trust region. But then  $t_k^E < 1$  and we have to find another technique to relate the model decrease to  $\Delta_k^2$ .

As in Section 6.3.3, a possible alternative to minimizing a quadratic model is to use an Armijo-like condition along a direction of negative curvature. More precisely, we continue to base our determination of an approximate eigenpoint on an approximate eigenvector  $u_k$  corresponding to the negative eigenvalue  $\tau_k$ ; that is, we still require

$$\langle u_k, g_k \rangle \leq 0, \quad \|u_k\|_k = \Delta_k, \quad \text{and} \quad \langle u_k, \nabla_{xx} m_k(x_k) u_k \rangle \leq \kappa_{\text{snc}} \tau_k [\nu_k^E]^2 \Delta_k^2, \quad (6.6.7)$$

but we now need to replace (6.6.3) by a more complicated but tractable condition. We first define

$$t_j = \kappa_{\text{bek}}^j \quad \text{and} \quad x_k(j) = x_k + t_j u_k \quad (6.6.8)$$

for  $j \geq 0$  and then determine the smallest nonnegative integer  $j = j_e$  such that

$$m_k(x_k(j)) \leq m_k(x_k) + \kappa_{\text{ubc}} \tau_k t_j^2 \|u_k\|^2, \quad (6.6.9)$$

where we have reused  $\kappa_{\text{bek}}$  and where  $\kappa_{\text{ubc}} \in (0, \frac{1}{2} \kappa_{\text{snc}})$  is a constant.<sup>71</sup> We may now define the *approximate eigenpoint* to be

---


$$t_k^{\text{AE}} = t_{j_e} \quad \text{and} \quad x_k^{\text{AE}} = x_k(t_k^{\text{AE}}) = x_k + t_k^{\text{AE}} u_k.$$

<sup>71</sup>“ubc” stands for “upper bound on the curvature”.

We then require, as in the previous section, that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{nmd}}(m_k(x_k) - \min[m_k(x_k^{\text{C}}), m_k(x_k^{\text{AE}})]). \quad (6.6.10)$$

Figure 6.6.3 illustrates the two possible cases that may arise when determining the approximate eigenpoint by such a backtracking procedure. In order to prove that the above conditions are coherent, we start by showing that  $t_{j_e}$  is well defined, in the sense that it is always finite. We simultaneously analyse the model reduction at  $x_k^{\text{AE}}$  compared to  $x_k$ .

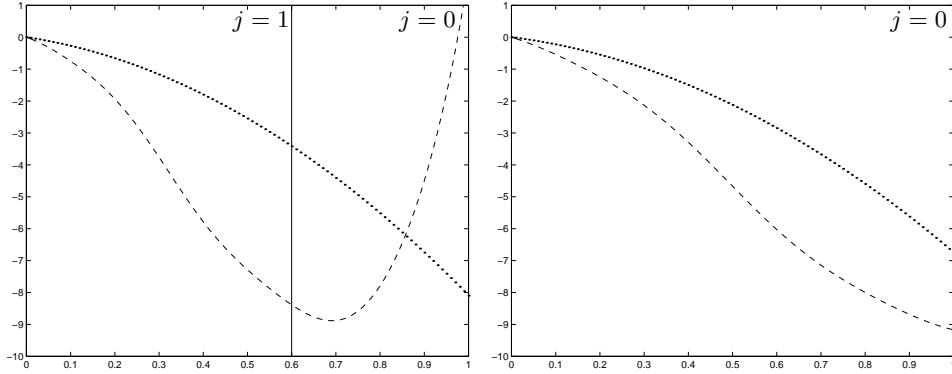


Figure 6.6.3: The two possibilities when determining the approximate eigenpoint by backtracking. In both pictures, the model is nonconvex and plotted with a dashed line, while the quadratic function on the right-hand side of (6.6.9) is plotted with a dotted line. In the picture on the left, (6.6.9) fails for  $j = 0$  ( $t = 1$ ) and is satisfied for  $j_e = 1$  ( $t = \kappa_{\text{bck}} = 0.6$ ). In the picture on the right, (6.6.9) holds for  $j_e = 0$  ( $t = 1$ ).

**Theorem 6.6.2** Suppose AM.1 holds, that  $\nabla_{xx}m_k(x_k)$  has a negative eigenvalue  $\tau_k$ , and that  $u_k$  satisfies the conditions (6.6.7). Suppose furthermore that the Hessian of the model is Lipschitz continuous with constant<sup>72</sup>  $\kappa_{\text{lch}} > 0$ , that is,

$$\|\nabla_{xx}m_k(x) - \nabla_{xx}m_k(y)\| \leq \kappa_{\text{lch}}\|x - y\| \quad (6.6.11)$$

for all  $x, y \in \mathcal{B}_k$ . Then we have that  $t_{j_e}$  is well defined and

$$m_k(x_k) - m_k(x_k^{\text{AE}}) \geq -\kappa_{\text{sod}}\tau_k \min[\tau_k^2, [\nu_k^{\text{E}}]^2 \Delta_k^2], \quad (6.6.12)$$

where  $\kappa_{\text{sod}} > 0$  is a constant.<sup>73</sup>

**Proof.** Consider first the case where (6.6.9) is violated for some  $j \geq 0$ , which

<sup>72</sup>“lch” stands for “Lipschitz constant for the model’s Hessian”.

<sup>73</sup>“sod” stands for “second-order decrease”.

implies that, for that  $j$ ,

$$m_k(x_k) - m_k(x_k(j)) < -\kappa_{\text{ubc}} \tau_k t_j^2 \|u_k\|^2 = -\kappa_{\text{ubc}} \tau_k t_j^2 [\nu_k^{\text{E}}]^2 \Delta_k^2, \quad (6.6.13)$$

where we have used the second part of (6.6.7) to derive the last equality. Applying the mean value theorem on the left-hand side on this inequality, we obtain, because of (6.6.7), that

$$\begin{aligned} m_k(x_k) - m_k(x_k(j)) &= -t_j \langle g_k, u_k \rangle - \frac{1}{2} t_j^2 \langle u_k, \nabla_{xx} m_k(\xi_j) u_k \rangle \\ &\geq -\frac{1}{2} t_j^2 \langle u_k, \nabla_{xx} m_k(\xi_j) u_k \rangle \end{aligned} \quad (6.6.14)$$

for some  $\xi_j$  in the segment  $[x_k, x_k + t_j u_k]$ . This is to say that

$$\begin{aligned} 2\kappa_{\text{ubc}} \tau_k [\nu_k^{\text{E}}]^2 \Delta_k^2 &\leq \langle u_k, \nabla_{xx} m_k(\xi_j) u_k \rangle \\ &= \langle u_k, \nabla_{xx} m_k(x_k) u_k \rangle + \langle u_k, [\nabla_{xx} m_k(\xi_j) - \nabla_{xx} m_k(x_k)] u_k \rangle \\ &\leq \kappa_{\text{snc}} \tau_k [\nu_k^{\text{E}}]^2 \Delta_k^2 + [\nu_k^{\text{E}}]^2 \Delta_k^2 \|\nabla_{xx} m_k(\xi_j) - \nabla_{xx} m_k(x_k)\| \\ &\leq \kappa_{\text{snc}} \tau_k [\nu_k^{\text{E}}]^2 \Delta_k^2 + \kappa_{\text{lch}} t_j [\nu_k^{\text{E}}]^3 \Delta_k^3, \end{aligned}$$

where we have successively used (6.6.14), (6.6.13), (6.6.7), the Cauchy–Schwarz inequality, (6.6.2), (6.6.11), and the bound  $\|\xi_j - x_k\| \leq t_j \|u_k\| = t_j \nu_k^{\text{E}} \Delta_k$ . As a consequence,

$$t_j \nu_k^{\text{E}} \Delta_k \geq \frac{2\kappa_{\text{ubc}} - \kappa_{\text{snc}}}{\kappa_{\text{lch}}} \tau_k > 0, \quad (6.6.15)$$

where the last inequality results from the bound  $\kappa_{\text{ubc}} < \kappa_{\text{snc}}/2$ . But  $t_j$  tends to zero when  $j$  tends to infinity, and thus (6.6.15) is impossible for  $j$  sufficiently large. This in turn implies that (6.6.13) must be false for  $j$  large enough, that is, that there must exist a finite  $j_e$  such that (6.6.9) holds and  $j_e$  is well defined. Moreover, (6.6.9) implies that

$$m_k(x_k) - m_k(x_k(j_e)) \geq -\kappa_{\text{ubc}} \tau_k t_{j_e}^2 [\nu_k^{\text{E}}]^2 \Delta_k^2. \quad (6.6.16)$$

We thus deduce from

$$t_{j_e} = \kappa_{\text{bck}} t_{j_e-1}$$

and the fact that (6.6.16) and thus (6.6.15) hold for  $j = j_e - 1$  that

$$m_k(x_k) - m_k(x_k(j_e)) = m_k(x_k) - m_k(x_k^{\text{AE}}) \geq -\kappa_{\text{ubc}} \kappa_{\text{bck}}^2 \left[ \frac{2\kappa_{\text{ubc}} - \kappa_{\text{snc}}}{\kappa_{\text{lch}}} \right]^2 \tau_k^3. \quad (6.6.17)$$

We now turn to the case where (6.6.9) holds for all  $j \geq 0$ , that is,  $j_e = 0$ . In that case,  $t_k^{\text{AE}} = 1$  and  $x_k^{\text{AE}}$  lies on the boundary of the trust region, and we deduce from (6.6.9) that

$$m_k(x_k) - m_k(x_k^{\text{AE}}) \geq -\kappa_{\text{ubc}} \tau_k [\nu_k^{\text{E}}]^2 \Delta_k^2. \quad (6.6.18)$$

Now combining (6.6.17) and (6.6.18) yields (6.6.12) with

$$\kappa_{\text{sod}} \stackrel{\text{def}}{=} \kappa_{\text{ubc}} \min \left( \kappa_{\text{bck}}^2 \left[ \frac{2\kappa_{\text{ubc}} - \kappa_{\text{snc}}}{\kappa_{\text{lch}}} \right]^2, 1 \right). \quad \square$$

Combining (6.6.10), (6.3.19), and (6.6.12), we obtain that

$$\begin{aligned} m_k(x_k) - m_k(x_k + s_k) \\ \geq \kappa_{\text{nmd}} \max \left\{ \kappa_{\text{dcp}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \nu_k^C \Delta_k \right], -\kappa_{\text{sod}} \tau_k \min[\tau_k^2, [\nu_k^E]^2 \Delta_k^2] \right\} \end{aligned} \quad (6.6.19)$$

when we use backtracking along an (approximate) eigenvector to determine the approximate eigenpoint. As for the model decrease at the approximate Cauchy point, we note that condition (6.6.19) obviously holds if the step  $s_k$  is computed to ensure that  $x_k + s_k$  is an approximation of the model minimizer  $x_k^M$ , in the sense that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{amm}} [m_k(x_k) - m_k(x_k^M)],$$

as this last condition guarantees that the model reduction obtained at  $x_k + s_k$  is at least a fraction of that obtained at  $x_k^M$ , which is itself at least equal to the largest of the reductions at the approximate Cauchy and eigenpoints, because of the bound

$$m_k(x_k^M) \leq \min[m_k(x_k^{AC}), m_k(x_k^{AE})].$$

### 6.6.3 Convergence Theorems

From the developments of the two preceding subsections, we see from (6.6.6) and (6.6.19) that the true and approximate eigenpoints yield essentially the same model reduction. As was the case for the Cauchy point in Section 6.3.4, we abandon the distinction between  $x_k^E$  and  $x_k^{AE}$ , the true and approximate eigenpoints, and only retain the notation and name of the former.

We also deduce from (6.6.6), (6.6.11), (6.6.19), and the fact that

$$\nu_k^E \geq \frac{1}{\kappa_{\text{une}}} > 0$$

for all  $k$  because of AN.1 that the following additional assumptions on Algorithm BTR are reasonable.

**AM.6** The Hessian of the model is Lipschitz continuous with constant  $\kappa_{\text{lch}}$  for all  $k$ ; that is, there exists a constant  $\kappa_{\text{lch}} > 0$  such that

$$\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\| \leq \kappa_{\text{lch}} \|x - y\|$$

for all  $x, y \in \mathcal{B}_k$ .

**AA.2** If  $\tau_k = \lambda_{\min}[\nabla_{xx} m_k(x_k)] < 0$ , then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min[\tau_k^2, \Delta_k^2]$$

for some constant  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$ .

Assumption AM.6 merely ensures that it is reasonable to impose AA.2, as indicated in Theorem 6.6.2, but is not used explicitly in what follows. Assumption AA.2 says that, if negative curvature appears in the model, if a first-order critical point is approached and the second-order terms of the model appear to be relevant,<sup>74</sup> then this negative curvature will be exploited by the calculation of the trust-region step. Once this is guaranteed, we may then proceed to examine the nature of the critical points to which Algorithm BTR is converging. In our new context, we may reformulate Theorem 6.3.4 as follows.

**Theorem 6.6.3** Suppose that AM.3, AM.5, AA.1, and AA.2 hold and that  $x_k$  is not second-order critical. Then  $m_k(x_k + s_k) < m_k(x_k)$  and  $s_k \neq 0$ .

**Proof.** As for Theorem 6.3.4, the conclusion immediately follows from the four stated assumptions and the observation that  $\Delta_k > 0$  for all  $k$ .  $\square$

We now start our convergence analysis by proving a first asymptotic result, stating that the objective must be convex in the neighbourhood of at least a subsequence of iterates.

**Theorem 6.6.4** Suppose that AF.1–AF.3, AN.1, AM.1–AM.6, AA.1, and AA.2 hold. Then

$$\limsup_{k \rightarrow \infty} \lambda_{\min}[\nabla_{xx} f(x_k)] \geq 0.$$

**Proof.** Assume, for the purpose of deriving a contradiction, that there is a constant  $\lambda_* < 0$  such that, for all  $k$ ,

$$\lambda_{\min}[\nabla_{xx} f(x_k)] \leq \lambda_*. \quad (6.6.20)$$

Then, since Theorem 6.4.6 ensures that  $\lim_{k \rightarrow \infty} \|g_k\| = 0$ , AM.5 then gives that, for  $k$  sufficiently large,

$$\lambda_{\min}[\nabla_{xx} m_k(x_k)] \leq \frac{1}{2}\lambda_*.$$

Using the same limit and AA.2, we then obtain that, for  $k$  sufficiently large,

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}\kappa_{\text{sod}}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \Delta_k^2]. \quad (6.6.21)$$

We now consider the ratio  $\rho_k$  of achieved versus predicted reduction when  $\Delta_k \leq \frac{1}{2}|\lambda_*|$ , and deduce from Lemma 6.5.3 applied to the complete sequence and the bound  $\|s_k\| \leq \nu_k^s \Delta_k \leq \kappa_{\text{une}} \Delta_k$  that there exists a  $k_0$  and a  $\delta_1 \leq \frac{1}{2}|\lambda_*|$  such that

$$\rho_k \geq \eta_2 \text{ for all } k \geq k_0 \text{ such that } \Delta_k \leq \delta_1.$$

---

<sup>74</sup>In the sense that the eigenpoint is below the Cauchy point.

Thus, each iteration  $k$  such that these two conditions hold must be very successful and the mechanism of the algorithm then ensures that  $\Delta_{k+1} \geq \Delta_k$ . As a consequence, we obtain that, for all  $j \geq 0$ ,

$$\Delta_{k_0+j} \geq \min[\gamma_1 \delta_1, \Delta_{k_0}] \stackrel{\text{def}}{=} \delta_2. \quad (6.6.22)$$

Combining (6.6.21) and this lower bound, we deduce that

$$f(x_{k_0+j}) - f(x_{k_0+j+1}) \geq \frac{1}{2} \eta_1 \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_2^2] > 0 \quad (6.6.23)$$

whenever iteration  $k_0 + j$  is successful. If there are infinitely many successful iterations after iteration  $k_0$ , then (6.6.23) contradicts AF.2. Thus all iterations must be unsuccessful for  $k$  large enough, and the mechanism of the algorithm then ensures that  $\Delta_k$  converges to zero. But this is again impossible because of (6.6.22). Hence (6.6.20) cannot hold and the proof of the theorem is complete.  $\square$

Observe that this theorem does not make the assumption that the sequence of iterates has limit points. For instance, if we consider the problem of minimizing  $e^{-x}$  for  $x \in \mathbb{R}$ , the sequence of iterates will diverge to plus infinity without reaching a limit point, but the theorem still applies. If a limit point exists, we still need the subsequence converging to this limit point to contain a further subsequence along which the corresponding objective function is asymptotically locally convex to obtain that the limit point is second-order critical. This is ensured in the following result, where we use the additional assumption

**AI.1** all iterates  $\{x_k\}$  lie within a closed, bounded domain  $\Omega$

to ensure the existence of limit points.

**Theorem 6.6.5** Suppose that AF.1–AF.3, AN.1, AM.1–AM.6, AA.1, AA.2, and AI.1 hold. Then there exists at least one limit point  $x_*$  of  $\{x_k\}$  which is second-order critical.

**Proof.** Theorem 6.6.4 ensures that there is a subsequence of iterates  $\{x_{k_i}\}$  such that

$$\lim_{i \rightarrow \infty} \lambda_{\min}[\nabla_{xx} f(x_{k_i})] \geq 0.$$

Since this sequence remains in a compact domain, it must have a limit point  $x_*$ . We then deduce that

$$\lambda_{\min}[\nabla_{xx} f(x_*)] \geq 0 \text{ and } \nabla_x f(x_*) = 0,$$

where the second equality follows from Theorem 6.4.6.  $\square$

To obtain further useful results on second-order critical points, we prove the following technical lemma.

**Lemma 6.6.6** Suppose that AF.1–AF.3, AN.1, AM.1–AM.6, AA.1, and AA.2 hold and that  $x_*$  is a limit point of the sequence of iterates produced by Algorithm BTR such that  $\lambda_* = \lambda_{\min}[\nabla_{xx}f(x_*)] < 0$ . Then we have that

$$\lim_{i \rightarrow \infty} \Delta_{k_i} = 0 \quad (6.6.24)$$

and  $\nabla_{xx}m_{k_i}(x_{k_i})$  has an eigenvalue

$$\tau_{k_i} \leq \frac{1}{2}\lambda_* \quad (6.6.25)$$

for  $i$  sufficiently large in every subsequence  $\{x_{k_i}\}$  of iterates converging to  $x_*$ .

**Proof.** Theorem 6.4.6 ensures that  $\|g_{k_i}\|$  converges to zero, which, combined with AM.5, gives that

$$\lim_{k \rightarrow \infty} \|\nabla_{xx}f(x_{k_i}) - \nabla_{xx}m_{k_i}(x_{k_i})\| = 0.$$

The continuity of the Hessian of the objective function, AF.1, then implies that  $\nabla_{xx}f(x_{k_i})$  has an eigenvalue at most equal to  $\lambda_*/2$  for  $i$  sufficiently large. Hence we may apply AM.5 and deduce that, for sufficiently large  $i$ ,  $\nabla_{xx}m_{k_i}(x_{k_i})$  has a negative eigenvalue  $\tau_{k_i}$  such that

$$\tau_{k_i} \leq \frac{1}{2}\lambda_*, \quad (6.6.26)$$

which is (6.6.25). In order to prove (6.6.24), we first consider the case where there are only a finite number of successful iterations. In this case, the mechanism of Algorithm BTR ensures that

$$\lim_{k \rightarrow \infty} \Delta_k = 0,$$

and thus (6.6.24) immediately follows. Assume now that there are an infinite number of successful iterations, and also suppose, for the purpose of deriving a contradiction, that there exists a subsequence  $\{x_{k_i}\}$  converging to  $x_*$  and an  $\epsilon \in (0, 1)$  such that

$$\Delta_{k_i} \geq \epsilon \quad (6.6.27)$$

for all  $i$ . As previously, we may now, without loss of generality, restrict our attention to successful iterates only, which gives that  $\{k_i\} \subseteq \mathcal{S}$ . Observe first that

$$\begin{aligned} -\tau_{k_i} \min[\tau_{k_i}^2, \Delta_{k_i}^2] &\geq \frac{1}{2}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \Delta_{k_i}^2] \\ &\geq \frac{1}{2}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \epsilon^2] \\ &\stackrel{\text{def}}{=} \delta, \end{aligned} \quad (6.6.28)$$

where we successively used (6.6.26) and (6.6.27). We may then apply AA.2 and deduce that, for  $i$  sufficiently large,

$$m_k(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{\text{sod}} \delta.$$

Since  $k_i \in \mathcal{S}$ , we have that, for  $i$  sufficiently large,

$$f(x_{k_i}) - f(x_{k_i+1}) \geq \eta_1 \delta \kappa_{\text{sod}}, \quad (6.6.29)$$

which is impossible since it would imply that  $f(x)$  is unbounded below, in contradiction with AF.2. As a consequence, we obtain that  $\Delta_{k_i}$  converges to zero for every subsequence of iterates converging to  $x_*$ , as requested.  $\square$

We may now prove a variant of Theorem 6.6.5 of more direct practical interest, because it covers the frequently occurring case where limit points are isolated and does not demand a priori that the sequence of iterates remains in a compact set.

**Theorem 6.6.7** Suppose that AF.1–AF.3, AN.1, AM.1–AM.6, AA.1, and AA.2 hold, and also that  $x_*$  is an isolated limit point of the sequence of iterates  $\{x_k\}$  produced by Algorithm BTR. Then  $x_*$  is a second-order critical point.

**Proof.** Let  $x_*$  be an isolated limit point of the sequence of iterates and  $\{x_{k_i}\}$  a subsequence of successful iterates converging to  $x_*$ . Assume furthermore, for the purpose of establishing a contradiction, that  $\nabla_{xx} f(x_*)$  has a negative eigenvalue  $\lambda_*$ . Then we may apply Lemma 6.6.6 and deduce that

$$\lim_{i \rightarrow \infty} \Delta_{k_i} = 0. \quad (6.6.30)$$

But, since  $x_*$  is isolated, there must exist a  $\delta > 0$  such that any other limit point of the sequence  $\{x_k\}$  is at a distance at least  $\delta$  from  $x_*$ . Moreover, we have that, for each  $x_k$  with  $k$  sufficiently large, either

$$\|x_k - x_*\| \leq \frac{1}{8}\delta \text{ or } \|x_k - x_*\| \geq \frac{1}{2}\delta. \quad (6.6.31)$$

In particular,

$$\|x_{k_i} - x_*\| \leq \frac{1}{8}\delta$$

for  $i$  large enough. Combining this last bound, the inequalities  $\|s_{k_i}\| \leq \nu_{k_i}^S \Delta_{k_i}$ , and  $\nu_{k_i}^S \leq \kappa_{\text{une}}$  (because of AN.1 and the limit (6.6.30)), we see that

$$\|x_{k_i+1} - x_*\| \leq \|x_{k_i} - x_*\| + \|s_{k_i}\| \leq \|x_{k_i} - x_*\| + \kappa_{\text{une}} \Delta_{k_i} \leq \frac{1}{8}\delta + \frac{1}{8}\delta = \frac{1}{4}\delta$$

for  $i$  sufficiently large. As a consequence, (6.6.31) implies that

$$\|x_{k_i+1} - x_*\| \leq \frac{1}{8}\delta$$

for  $i$  sufficiently large. Applying this argument repeatedly, we obtain that the complete sequence  $\{x_k\}$  converges to  $x_*$ . The limit (6.6.30) then ensures that

$$\lim_{k \rightarrow \infty} \Delta_k = 0 \quad (6.6.32)$$

and thus that

$$\lim_{k \rightarrow \infty} \|s_k\| = 0. \quad (6.6.33)$$

Now, AF.1 gives that, for  $k$  sufficiently large,  $\nabla_{xx}f(x_k)$  must have an eigenvalue  $\lambda_k \leq \lambda_*/2$ , where  $\lambda_*$  is as in (6.6.20) and thus, by AM.5 and the convergence of  $\|g_k\|$  to zero, that  $\nabla_{xx}m_k(x_k)$  has an eigenvalue  $\tau_k \leq \kappa_{\text{snc}}\lambda_*/2$  for  $k$  large enough (recall  $\kappa_{\text{snc}} \in (0, 1]$ ). As a consequence, AA.2 then implies that, for  $k$  sufficiently large,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}}|\tau_k| \min[\tau_k^2, \Delta_k^2] \geq \frac{1}{2}\kappa_{\text{sod}}\kappa_{\text{snc}}|\lambda_*|\Delta_k^2 > 0, \quad (6.6.34)$$

where the penultimate inequality results from (6.6.32). Again, (6.6.33), (6.6.34), and the bound  $\|s_k\| \leq \kappa_{\text{une}}\Delta_k$  allow us to apply Lemma 6.5.4 with

$$\kappa_{\text{mqd}} = \frac{\kappa_{\text{sod}}\kappa_{\text{snc}}|\lambda_*|}{2\kappa_{\text{une}}^2},$$

and we obtain that all iterations are eventually successful and  $\Delta_k$  is bounded away from zero. But this contradicts (6.6.32), and we therefore deduce that our initial assumption is false, which concludes our proof.  $\square$

Observe that neither Theorem 6.6.5 nor Theorem 6.6.7 ensures that the complete sequence converges or, more importantly, that the trust-region radius  $\Delta_k$  is bounded away from zero, which is highly desirable if we wish to ensure a fast rate of convergence. Indeed, the situation where there is a curve of limit points (all first-order critical, because of Theorem 6.4.6) is not excluded. In such a case, it might be that the Hessian of the objective function has a nontrivial null-space at each such limit point and the values of the objective function (and, consequently, of  $\rho_k$  and  $\Delta_k$ ) are no longer adequately described, in this null-space, simply by first- and second-order information. We would thus need assumptions involving higher order derivatives of  $f$  and  $m_k$  to characterize the behaviour of Algorithm BTR in these situations. However, if  $\nabla_{xx}f(x_*)$  happens to be not simply positive semidefinite at the limit point(s) considered in Theorems 6.6.5 or 6.6.7, but actually positive definite, then Theorem 6.5.5 applies and its stronger convergence results are applicable.

In order to obtain an even better convergence result, we now introduce a very slight modification of Algorithm BTR. The main idea is to strengthen the conditions governing the trust-region radius update, so that this radius actually increases following very successful iterations. Specifically, we will require the following.

**AA.3** If  $\rho_k \geq \eta_2$  and  $\Delta_k \leq \Delta_{\max}$ , then

$$\Delta_{k+1} \in [\gamma_3\Delta_k, \gamma_4\Delta_k]$$

for some  $\gamma_4 \geq \gamma_3 > 1$  and some  $\Delta_{\max} > 0$ .

Note that this additional requirement on Algorithm BTR is very simple and intuitively appealing, because it seems reasonable to expand the trust-region radius whenever the iteration has been very successful, at least when it is not already too large.<sup>75</sup> It is remarkable that such a small modification allows us to prove the following strong convergence result.

**Theorem 6.6.8** Suppose that AF.1–AF.3, AN.1, AM.1–AM.6, and AA.1–AA.3 hold and let  $x_*$  be any limit point of the sequence of iterates. Then  $x_*$  is a second-order critical point.

**Proof.** The beginning of this proof is very similar to that of Theorem 6.6.4. We first note that

$$\lim_{k \rightarrow \infty} \|g_k\| = 0 \quad (6.6.35)$$

because of Theorem 6.4.6. Assume now, for the purpose of deriving a contradiction, that

$$\lambda_* = \lambda_{\min}[\nabla_{xx} f(x_*)] < 0. \quad (6.6.36)$$

Then AM.5 ensures that there exist a  $k_0$  and a  $\delta > 0$  such that

$$\lambda_{\min}[\nabla_{xx} m_k(x_k)] \leq \frac{1}{2}\lambda_* < 0 \quad (6.6.37)$$

for every  $k \geq k_0$  for which

$$x_k \in \mathcal{B}(x_*, \delta) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid \|x - x_*\| \leq \delta\}.$$

Let  $\mathcal{K} = \{k \geq k_0 \mid x_k \in \mathcal{B}(x_*, \delta)\}$ . Using this definition, AA.2, and (6.6.35), we then obtain that, for  $k \in \mathcal{K}$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}\kappa_{\text{sod}}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \Delta_k^2]. \quad (6.6.38)$$

We may then again consider the ratio  $\rho_k$  of achieved versus predicted reduction for  $\Delta_k \leq \frac{1}{2}|\lambda_*|$  and deduce from Lemma 6.5.3 applied to the subsequence  $\mathcal{K}$  and the bound  $\|s_k\| \leq \nu_k^S \Delta_k \leq \kappa_{\text{une}} \Delta_k$  that there exists a  $k_0$  and a  $\delta_1 \leq \frac{1}{2}|\lambda_*|$  such that

$$\rho_k \geq \eta_2 \geq \eta_1 \text{ for all } k \geq k_0 \text{ such that } x_k \in \mathcal{B}(x_*, \delta) \text{ and } \Delta_k \leq \delta_1. \quad (6.6.39)$$

Therefore, each iteration for which these two conditions hold must be very successful.

Now let  $x_\ell$  ( $\ell \geq k_0$ ) be an iterate such that  $\|x_\ell - x_*\| \leq \frac{1}{2}\delta$  and consider the sequence  $\{x_{\ell+j}\}$  ( $j \geq 0$ ). Either this sequence remains within  $\mathcal{B}(x_*, \delta)$  or it eventually leaves it. In the first case, that is, if it stays within  $\mathcal{B}(x_*, \delta)$ , then (6.6.39) ensures that

---

<sup>75</sup>If  $\Delta_k > \Delta_{\max}$  and  $\rho_k \geq \eta_2$ , we merely assume that  $\Delta_{k+1} \geq \Delta_k$ , as stated by (6.1.5) (p. 116).

each iteration must be very successful if the trust-region radius becomes as small as  $\delta_1$ , and thus

$$\Delta_{\ell+j} \geq \min[\gamma_1 \delta_1, \Delta_\ell] \stackrel{\text{def}}{=} \delta_2 \quad (6.6.40)$$

for  $j \geq 0$ . Combining (6.6.38) and this lower bound, we obtain that

$$f(x_{\ell+j}) - f(x_{\ell+j+1}) \geq \frac{1}{2} \eta_1 \kappa_{\text{sod}} |\lambda_*| \min\left[\frac{1}{4} \lambda_*^2, \delta_2^2\right] > 0 \quad (6.6.41)$$

whenever iteration  $\ell + j$  is successful. If there are only finitely many successful iterations, the mechanism of the algorithm implies that the trust-region radius converges to zero, which is impossible because of (6.6.40). Hence there must be an infinite number of successful iterations. But (6.6.41) now contradicts AF.2. We thus conclude that the sequence  $\{x_{\ell+j}\}$  must leave  $\mathcal{B}(x_*, \delta)$  and the first of our two cases cannot happen.

In the second case, that is, if the sequence  $\{x_{\ell+j}\}$  leaves  $\mathcal{B}(x_*, \delta)$ , there must be a first successful iteration of index  $p \geq \ell$  such that  $x_\ell = x_p \neq x_{p+1}$ . If we let  $x_{q+1}$  be the first iterate outside  $\mathcal{B}(x_*, \delta)$ , we have that

$$\frac{\delta}{2} \leq \|x_{q+1} - x_\ell\| \leq \sum_{\substack{k=p \\ k \in \mathcal{S}}}^q \|x_{k+1} - x_k\| \leq \kappa_{\text{une}} \sum_{\substack{k=p \\ k \in \mathcal{S}}}^q \Delta_k, \quad (6.6.42)$$

where we have used AN.1 and (6.4.2) to derive the last inequality. Assume first that there exists a smallest integer  $j$  such that

$$\Delta_j > \min[\Delta_{\max}, \delta_1] \stackrel{\text{def}}{=} \delta_4 \quad \text{and} \quad p \leq j \leq q. \quad (6.6.43)$$

If iteration  $j$  is successful, we obtain from (6.6.38) that

$$f(x_p) - f(x_{q+1}) \geq f(x_j) - f(x_{j+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min\left[\frac{1}{4} \lambda_*^2, \delta_4^2\right] \stackrel{\text{def}}{=} \delta_3 > 0. \quad (6.6.44)$$

If iteration  $j$  is unsuccessful, then  $j > p$ , because  $p \in \mathcal{S}$ , and AA.3 then ensures that

$$\delta_4 < \Delta_j \leq \gamma_4 \Delta_{j-1}. \quad (6.6.45)$$

Moreover, since  $j$  is the smallest such that (6.6.43) holds, we have that  $\Delta_{j-1} < \Delta_j$ , and therefore that  $j-1 \in \mathcal{S}$ . As a consequence, we deduce from (6.6.45) and (6.6.38) that

$$f(x_p) - f(x_{q+1}) \geq f(x_{j-1}) - f(x_j) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min\left[\frac{1}{4} \lambda_*^2, \left(\frac{\delta_4}{\gamma_4}\right)^2\right] \stackrel{\text{def}}{=} \delta_5 > 0. \quad (6.6.46)$$

On the other hand, if no  $j$  between  $p$  and  $q$  satisfies (6.6.43), then (6.6.39) implies that all iterations from  $p$  to  $q$  are very successful, and we obtain from AA.3 that the right-hand side of (6.6.42) may be bounded by

$$\sum_{\substack{k=p \\ k \in \mathcal{S}}}^q \Delta_k \leq \sum_{\substack{k=p \\ k \in \mathcal{S}}}^q \frac{\Delta_q}{\gamma_3^{q-k}} \leq \frac{\gamma_3}{\gamma_3 - 1} \Delta_q. \quad (6.6.47)$$

Combining (6.6.42) and (6.6.47), we obtain that

$$\Delta_q \geq \frac{(\gamma_3 - 1)\delta}{2\gamma_3\kappa_{\text{une}}} \stackrel{\text{def}}{=} \delta_6.$$

Furthermore, the definition of  $q$  implies that  $q \in \mathcal{S}$ , and therefore, using (6.6.38), that

$$f(x_p) - f(x_{q+1}) \geq f(x_q) - f(x_{q+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4}\lambda_*^2, \delta_6^2] \stackrel{\text{def}}{=} \delta_7 > 0. \quad (6.6.48)$$

Combining (6.6.44), (6.6.46), and (6.6.48), we obtain that

$$f(x_p) - f(x_{q+1}) \geq \min[\delta_3, \delta_5, \delta_7] > 0.$$

Our assumption that the objective function is bounded below, AF.2, then yields that the sequence  $\{x_{\ell+j}\}$  can only leave  $\mathcal{B}(x_*, \delta)$  a finite number of times. Since we have shown that it cannot remain within  $\mathcal{B}(x_*, \delta)$ , we obtain that

$$\|x_k - x_*\| \geq \frac{\delta}{2}$$

for all  $k$  sufficiently large, which in turn contradicts the assumption that  $x_*$  is a limit point of the sequence of iterates. Hence our assumption (6.6.36) must be false and the theorem is proved.  $\square$

This result implies that every limit point  $x_*$  of the sequence of iterates produced by the algorithm is second-order critical if AA.3 replaces the additional assumptions that the sequence is bounded (Theorem 6.6.5) or that  $x_*$  is isolated (Theorem 6.6.7).

## Notes and References for Section 6.6

The asymptotic local convexity of the objective for a subsequence of the iterates (Theorem 6.6.4) appeared in Conn, Gould, Orban, and Toint (2000) in the context of using the log-barrier function for nonconvex problems with linear equality and bound constraints, and was inspired by Shultz, Schnabel, and Byrd (1985). Convergence to a second-order critical point (Theorem 6.6.5) was proved independently by Fletcher (1980, p. 78) and Sorensen (1982a) for the case where  $s_k$  is determined from the model minimizer  $x_k^M$ . At the same time, Shultz, Schnabel, and Byrd (1985) introduced the explicit use of an eigenpoint, although the terminology is itself specific to our presentation. Knoth (1983) also considered the use of negative curvature in trust-region-based versions of Newton's method. Theorem 6.6.7 was proved by Sorensen (1982a) and Moré and Sorensen (1983), while Gay (1982) covered the case where convergence occurs to points at which the Hessian is singular. Finally, the observation that stronger convergence results could be obtained with AA.3 was first made by Shultz, Schnabel, and Byrd (1985). Their original formulation is expressed for an algorithm which is slightly different from Algorithm BTR, as it includes a version of the "internal doubling" technique (see Section 10.5.1). It does not include the algorithmic safeguard that  $\Delta_k \leq \Delta_{\max}$ , which we have introduced to preserve the computationally desirable requirement that the trust-region radius be bounded above. See also Yuan (2000).

Most of the preceding references restrict their analysis to the case where the model is quadratic. More general models were independently introduced by Carter (1986) and Toint (1988). Conic models, i.e., models of the form

$$m_k(x_k + s) = m_k(x_k) + \frac{\langle g_k, s \rangle}{1 + \langle h_k, s \rangle} + \frac{\langle s, H_k s \rangle}{2(1 + \langle h_k, s \rangle)^2},$$

have been considered by Di and Sun (1996), W. Sun (1996), Han and Han (1999), and Sun and Yuan (1998), while Ariyawansa and Lau (1992) discuss the use of trust-region techniques to determine the vector  $h_k$ . Models of an order higher than 2 (tensor models) are discussed in Hanson and Krogh (1992), Chow and Chen (1994), W. Sun (1996), Bouaricha (1997), Bouaricha and Schnabel (1997), Corradi (1997), and Bouaricha and Schnabel (1998, 1999). See also Alexandrov et al. (1998). The theorems of this section appear to be new generalizations of existing results in this extended framework.

The literature on convergence theory for trust-region methods contains further interesting contributions. For instance, we note the interpretation of the trust-region algorithm as a linearized implicit Euler method with adaptive timestep for ordinary differential equations, which allows Higham (1999) to discuss both global convergence and rate of convergence properties.

Finally, we have mentioned the importance of local convergence analysis, in which the rate of convergence of optimization algorithms is assessed. The reader is referred to Dennis and Schnabel (1983) for further details and references on this subject.

## 6.7 Trust-Region Scaling

### 6.7.1 Geometry and Scaling

As noted in Section 6.1, the choice of norm when defining the trust region

$$\mathcal{B}_k = \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k\}$$

determines the geometrical *shape* of the trust region. Furthermore, this shape determines the positions of the Cauchy point and eigenpoint, as well as that of the model minimizer. Figure 6.7.1 illustrates this.

It is not uncommon that problems arising in practice involve variables whose “typical values” (in the terminology of Dennis and Schnabel, 1983, Section 7.1) are of vastly different orders of magnitude. For instance, such a problem might be related to the design of an electrical RC circuit, in which resistances are expressed in ohms and capacitances in farads, the two standard units for such quantities. However, the circuit under study may involve resistances of values between  $10^3$  and  $10^6$  ohms and capacitances of the order of  $10^{-12}$  farads! If the optimization problem is expressed in ohms and farads, we see that small changes in the capacitance variables are likely to affect the value of the objective function much more than small changes in the resistance variables. In this case, we say that the problem is *badly scaled*. This may lead to severe numerical difficulties; for instance the contribution of the capacitance variables may be completely dwarfed by that of the resistance variables and thus numerically swamped when computing  $\|s_k\|^2$ . Indeed, if  $s_k = r_k + c_k$ , where  $r_k$  and  $c_k$ , respectively, represent

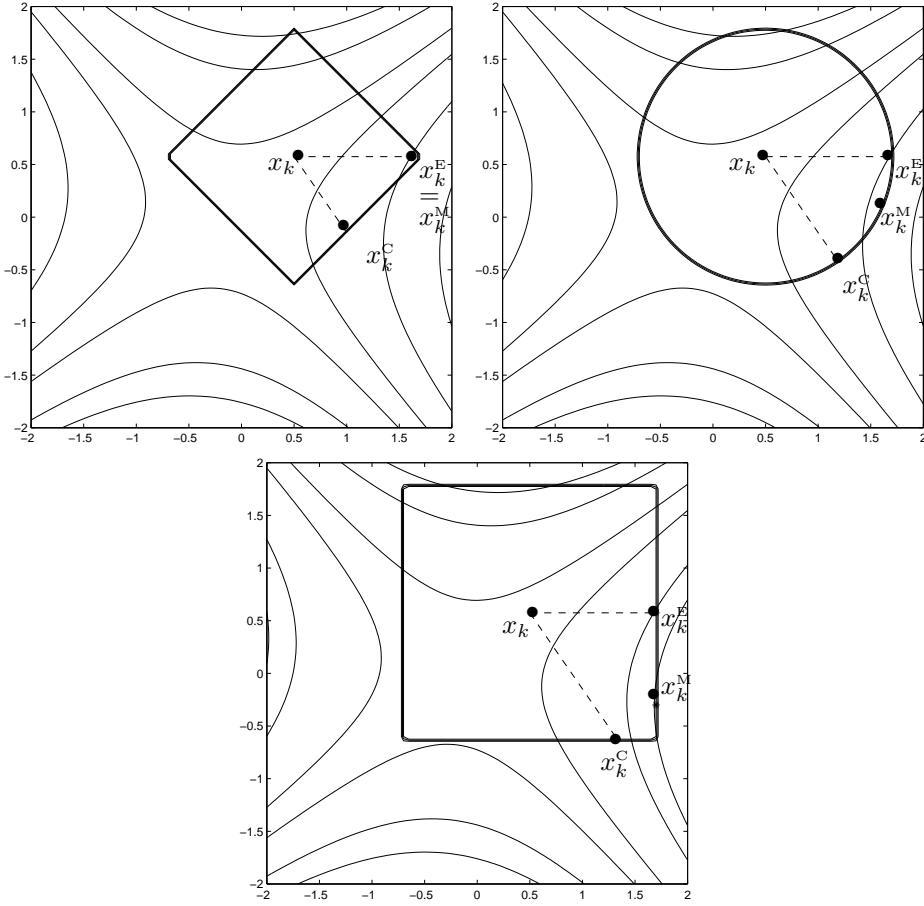


Figure 6.7.1: The trust region, Cauchy and eigenpoints, and model minimizers for a given function (as indicated by the contours) using the  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms.

the resistance and capacitance components of the step, we see that  $\|r_k\|^2 \approx 10^{12}$  and  $\|c_k\|^2 \approx 10^{-24}$ . If we assume we have 15 digits of accuracy on our computer, then  $\|s_k\|^2 = \|r_k\|^2$  and the capacitance information is lost! Furthermore this means that the trust-region radius cannot effectively limit  $\|c_k\|$ , making Algorithm BTR and its associated convergence theory of dubious practical value. It is thus of paramount importance to rescale the problem's variables to make their typical values of comparable magnitude, if at all possible. In our example, this would amount to performing a diagonal change of variables  $w = S^{-1}x$ , where the diagonal matrix  $S$  contains the “typical sizes” of the problem’s variables. Thus

$$S = \begin{pmatrix} 10^6 & 0 \\ 0 & 10^{-12} \end{pmatrix},$$

where we have grouped the variables associated with resistances first.

Of course, not every example is so extreme, and not every variable scaling matrix is diagonal. It may in particular depend on an eigenvalue analysis of the Hessian of

the objective function. We may even consider that, due to the nonlinearity of our application, the scaling must depend on the current iterate  $x_k$ ; we then obtain the general situation where we wish to use, at iteration  $k$ , a given nonsingular scaling matrix  $S_k$ , such that  $f(x_k + S_k w)$  is well scaled as a function of the new variable  $w$ . The matrix  $S_k$  may, for instance, be constructed via a suitable preconditioner for the matrix<sup>76</sup>  $H_k$ . Once  $w$  is determined by the relation

$$S_k w = s, \quad (6.7.1)$$

it is natural to see the trust-region problem in terms of  $w$ , which means that we would like to build a (scaled) model  $m_k^s$  such that

$$m_k^s(x_k + w) \approx f(x_k + S_k w) \stackrel{\text{def}}{=} f^s(w), \quad (6.7.2)$$

which we may trust in a (scaled) trust region defined as

$$\mathcal{B}_k^s = \{x_k + w \mid \|w\| \leq \Delta_k\}.$$

Assume, for a moment, that we choose  $m_k^s$  to be quadratic. The requirement (6.7.2) then imposes, in line with AM.2 and AM.3, that

$$m_k^s(x_k) = f(x_k), \quad g_k^s = \nabla_w m_k^s(x_k) = \nabla_w f^s(0) = S_k^T \nabla_x f(x_k), \quad (6.7.3)$$

and

$$H_k^s = \nabla_{ww} m_k^s(x_k) \approx \nabla_{ww} f^s(0) = S_k^T \nabla_{xx} f(x_k) S_k. \quad (6.7.4)$$

Thus we have that, as expected,

$$\begin{aligned} m_k^s(x_k + w) &= f(x_k) + \langle g_k^s, w \rangle + \frac{1}{2} \langle w, H_k^s w \rangle \\ &= f(x_k) + \langle S_k^T \nabla_x f(x_k), w \rangle + \frac{1}{2} \langle w, S_k^T H_k S_k w \rangle \\ &= f(x_k) + \langle \nabla_x f(x_k), S_k w \rangle + \frac{1}{2} \langle S_k w, H_k S_k w \rangle \\ &= f(x_k) + \langle \nabla_x f(x_k), s \rangle + \frac{1}{2} \langle s, H_k s \rangle \\ &= m_k(x_k + s), \end{aligned} \quad (6.7.5)$$

where  $H_k$  approximates  $\nabla_{xx} f(x_k)$ . Expressed in the scaled space, the trust-region subproblem is

$$\min_{\|w\| \leq \Delta_k} m_k^s(x_k + w), \quad (6.7.6)$$

which, in view of (6.7.5) and (6.7.1), can be expressed in terms of the problem's original variables as

$$\min_{\|S_k^{-1} s\| \leq \Delta_k} m_k(x_k + s). \quad (6.7.7)$$

We may then prefer to use (6.7.6) when actually computing the step, because of the better scaling properties of  $w$ . If we now consider models more general than quadratic, the situation is similar in that conditions (6.7.3) still apply, while (6.7.4) only provides some indication of how  $m_k^s$  should be chosen.

---

<sup>76</sup>See Section 5.1.6 for a method to handle preconditioners.

We also note that the trust region of (6.7.7) is given by

$$\mathcal{B}_k^S = \{s \in \mathbb{R}^n \mid \|s\|_k \leq \Delta_k\}, \quad (6.7.8)$$

where we have defined

$$\|s\|_k^2 \stackrel{\text{def}}{=} \langle S_k^{-1}s, S_k^{-1}s \rangle = \langle s, S_k^{-T}S_k^{-1}s \rangle, \quad (6.7.9)$$

which is an ellipsoidal norm with the shape of the ellipse being determined by the symmetric positive definite matrix  $S_k^{-T}S_k^{-1}$ . We thus conclude that (iteration-dependent) *rescaling of the variables can be interpreted as using a trust region defined by an (iteration-dependent) scaled norm (6.7.8)–(6.7.9)*.

Figure 6.7.2 illustrates an example where the variable corresponding to the hori-

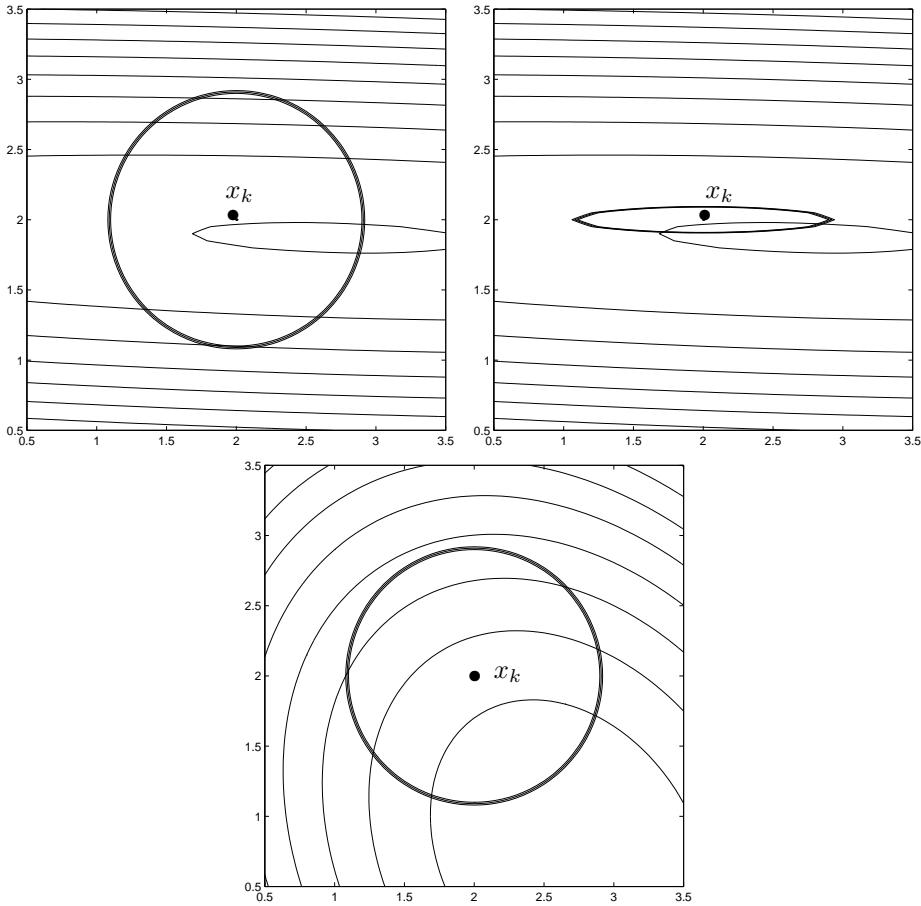


Figure 6.7.2: Top left: unscaled model with unscaled trust region (6.3.28); top right: scaled model and trust region, seen in the unscaled space (6.7.7); bottom: scaled model and trust region seen in the scaled space (6.7.6).

zontal axis is 10 times smaller<sup>77</sup> than that on the vertical axis, which is to say that

$$S_k = \begin{pmatrix} 1 & 0 \\ 0 & 10 \end{pmatrix}.$$

### 6.7.2 Uniformly Equivalent Norms Again

Of course, since the norm  $\|\cdot\|_k$  is another norm on  $\mathbb{R}^n$ , it is also equivalent to the  $\ell_2$  norm. However, this simple equivalence may not be enough for our purposes, mostly because we may be ignoring whether the different norms used at different iterations are coherent within the complete minimization process. This can be important, as is shown by the following two examples. Consider first the one-dimensional problem of minimizing  $x^2$ , using Newton's method<sup>78</sup> and a trust region of the form

$$\mathcal{B}_k = \{x \in \mathbf{R} \mid k! |x - x_k| \leq \Delta_k\}. \quad (6.7.10)$$

Note that  $\nu_k^C = 1/k!$ , which converges to zero. If we start from  $x_0 = 2e$ , where  $e = \exp(0)$ , and  $\Delta_0 = 1$ , it is easy to verify that all iterations are successful, we may choose  $\Delta_k = 1$  for all  $k$ , and the iterates are given by

$$x_k = 2e - \sum_{j=0}^k \frac{1}{j!}.$$

This immediately yields that

$$x_k \geq 2e - \sum_{j=0}^{\infty} \frac{1}{j!} = 2e - e = e,$$

which shows that convergence will never occur to the (unique) critical point of the objective function. This may be seen as a consequence of the fact that the steps shorten too quickly. But the opposite problem can also arise.

Consider the one-dimensional problem of minimizing,  $f(x) = x^4 - 10(x+1)^2$ , again using Newton's model and starting from  $x_0 = 0$  and  $\Delta_0 = 4$ . This problem is nonconvex but has a unique, well-defined minimizer at  $x_* \approx 2.63$ . Figure 6.7.3 shows the objective function and its quadratic model at the starting point.

If we choose to determine the step  $s_k$  by minimizing the model in the trust region defined by

$$\mathcal{B}_k = \{x \in \mathbf{R} \mid |x - x_k| \leq (k+1)! \Delta_k\},$$

we immediately see that  $x_0 + s_0 = x_0^M = x_0 + \Delta_0$ . Moreover,  $f(x_0 + s_0) > f(x_0)$  and iteration 0 is thus unsuccessful. We then choose  $\Delta_1 = \frac{1}{2}\Delta_0$ . Thus  $\mathcal{B}_1 = [-4, 4]$  again, and iteration 1 is thus again unsuccessful. A simple induction then shows that  $x_k + s_k = x_k^M = 4(\frac{1}{2})^k(k+1)!$  (which tends to infinity), and that all iterations are unsuccessful, preventing convergence to the solution. In this case, the effect of the

---

<sup>77</sup>A very modest difference of scale, which is chosen for a clearer picture.

<sup>78</sup>The model and the objective function coincide in this case, and the step  $s_k = x_k^M - x_k$  is used.

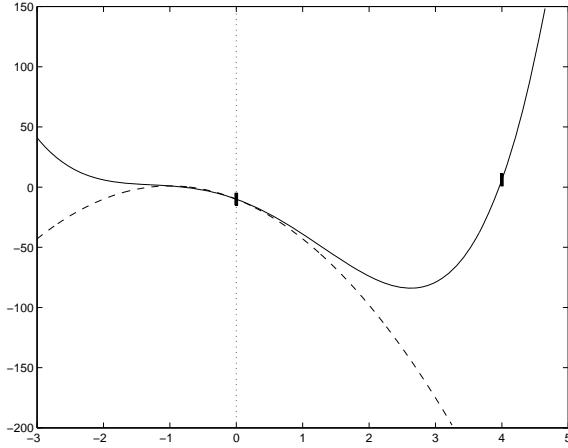


Figure 6.7.3: The function  $f(x) = x^4 - 10(x + 1)^2$  (solid) and its Newton model at  $x_0 = 0$  (dashed). The starting point and  $x_0 + s_0$  are ticked on the objective function curve.

iteration-dependent norm  $\|\cdot\|_k = |\cdot|/(k+1)!$  is to make the steps larger and larger, despite the unsuccessful nature of all iterations. In this case, we see that  $\nu_k^S = (k+1)!$ , which diverges to infinity. Thus we conclude from these two examples that, although each of the norms  $\|\cdot\|_k$  is individually equivalent to the  $\ell_2$  norm with equivalence constants  $\kappa_{n2k}$ , say, the fact that these equivalence constants are not bounded away from zero or infinity may cause Algorithm BTR to fail. In other words, convergence may not occur if subsequences of  $\{\nu_k^C\}$  or  $\{\nu_k^S\}$  are allowed to converge to zero or infinity, respectively. We therefore have to prevent that situation, which may be done by assuming uniform equivalence between all the norms we consider, which we already expressed in AN.1. An illustration of this concept in the framework of ellipsoidal norms of the form (6.7.9) is shown in Figure 6.7.4.

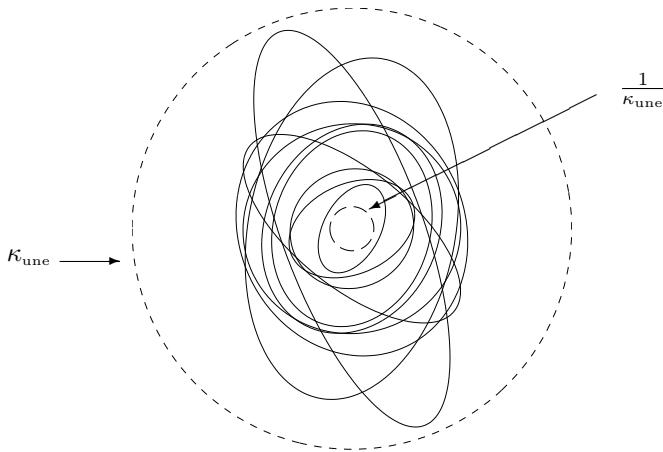


Figure 6.7.4: The shape of 10 trust regions defined using 10 uniformly equivalent ellipsoidal norms with  $\Delta_k = 1$ .

If we return to the case where  $\|\cdot\|_k$  reflects an iteration-dependent scaling of the variables, it is important to note that AN.1 is guaranteed if the scaling matrices  $S_k$  are “uniformly bounded above and below”, as is expressed by the following easy result.

**Theorem 6.7.1** Suppose that there exists a constant<sup>79</sup>  $\kappa_{\text{scb}} \geq 1$  such that, for all  $k$ ,

$$\frac{1}{\kappa_{\text{scb}}} \leq \sigma_{\min}[S_k] \leq \sigma_{\max}[S_k] \leq \kappa_{\text{scb}}.$$

Then AN.1 holds.

**Proof.** The inequalities

$$\|x\|_k = \|S_k^{-1}x\| \geq \sigma_{\min}[S_k^{-1}]\|x\| = \frac{1}{\sigma_{\max}[S_k]}\|x\| \geq \frac{1}{\kappa_{\text{scb}}}\|x\|$$

and

$$\|x\|_k = \|S_k^{-1}x\| \leq \|S_k^{-1}\|\|x\| \leq \sigma_{\max}[S_k^{-1}]\|x\| = \frac{1}{\sigma_{\min}[S_k]}\|x\| \leq \kappa_{\text{scb}}\|x\|$$

immediately imply the desired result.  $\square$

Clearly, choosing a suitable norm  $\|\cdot\|_k$  in Algorithm BTR is of interest if we wish to scale the problem’s variables or, more generally, modify the shape of the trust region itself.

We conclude by pointing out that our convergence theory for Algorithm BTR only uses AN.1 in a very specific way. In fact, we only need the two inequalities

$$\nu_k^C \geq \frac{1}{\kappa_{\text{une}}} \quad \text{and} \quad \nu_k^S \leq \kappa_{\text{une}} \tag{6.7.11}$$

to obtain convergence to first-order critical points, and

$$\nu_k^E \geq \frac{1}{\kappa_{\text{une}}} \tag{6.7.12}$$

for convergence to second-order ones, which shows that the scaling along the directions  $-g_k$ ,  $s_k$ , and  $u_k$  is all that matters. In theory, (6.7.11)–(6.7.12) is much less restrictive than assuming AN.1, and we will use this possible relaxation later.

## Notes and References for Section 6.7

To our knowledge, the first paper to explicitly mention scaling in trust-region algorithms is Moré (1978). Other early references include Dennis and Schnabel (1983), Moré and Sorensen (1984), and Gay (1984).

---

<sup>79</sup>“scb” stands for “scaling condition bound”.

# Chapter 7

---

---

## The Trust-Region Subproblem

---

---

### 7.1 The Solution of Trust-Region Subproblems

In the previous chapter, we have seen that the central computation in our basic trust-region algorithm is the calculation of a step  $s_k$  that sufficiently reduces the model  $m_k(x_k + s)$  and for which  $x_k + s_k \in \mathcal{B}_k$ . We have also observed that the Cauchy point gives such a step, as indeed do any other points within the trust region that give a reduction in the model which is a positive fraction of that obtained at the Cauchy point.

In this chapter, we consider a number of possibilities in the special, but frequent, case where the model we wish to employ is the quadratic

$$m_k(x_k + s) = f(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle, \quad (7.1.1)$$

where  $g_k$  is the gradient  $\nabla f(x_k)$  and  $H_k$  is a symmetric bounded approximation to  $\nabla_{xx} f(x_k)$ . There are really two obvious but extreme possibilities for  $s_k$ . The first is the step to the Cauchy point. The second is a step that makes  $m_k(x_k + s)$  as small as possible within  $\mathcal{B}_k$ . As in Section 6.3.4, we shall call a point that makes the model as small as possible the *model minimizer* and denote it by  $x_k^M$ .

Realistically, any point that gives a model value between these two extremes is acceptable in terms of the theory we have just studied. Practically, as we shall see, there are contrasting differences between the two possibilities. The Cauchy point is in general cheap to obtain, while computing the model minimizer requires the solution of a minimization problem and may therefore be expensive. Conversely, continued use of the Cauchy *step* (that is, the step to the Cauchy point) normally results in a slowly (Q-linearly) converging method, while an asymptotically fast (Q-superlinear or quadratic) rate of convergence is possible<sup>80</sup> with the step to the model minimizer. This then suggests that a compromise step between these two extremes may give us the computational efficiencies of the former and the convergence advantages of the latter.

---

<sup>80</sup>For instance, a model based upon a second-order Taylor approximation will normally give Q-quadratic convergence.

Before we start, there remains one further complication. Although, as we have seen, as long as the norms are uniformly equivalent (see AN.1), the choice of norm that defines the trust region is irrelevant to the ultimate convergence of the method, that choice may have a serious impact on the core computations. A user is spoilt for choice. However, in our experience, most people prefer the  $\ell_1$ ,  $\ell_2$ , or  $\ell_\infty$  norms, or scaled variants of these. But as we can see in Figure 7.1.1, the nature of the Cauchy point and model minimizer can be very different for these norms.

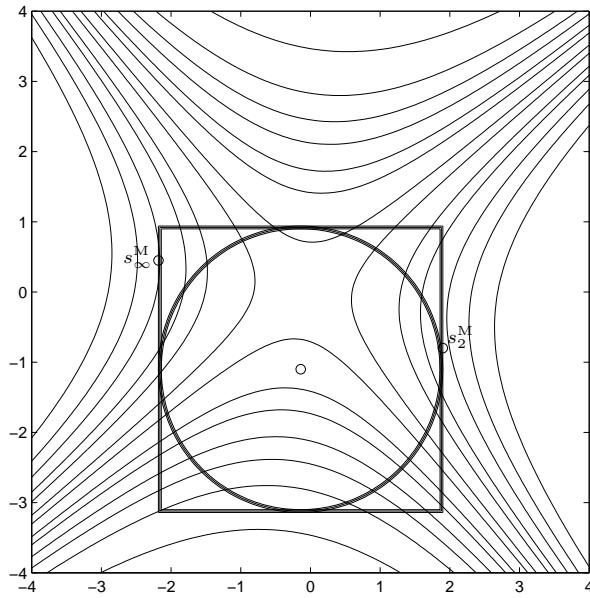


Figure 7.1.1:  $s_2^M$  and  $s_\infty^M$ , the  $\ell_2$  and the  $\ell_\infty$  model minimizers.

The  $\ell_1$  norm has not proved popular in trust-region methods, probably because it has the bad features of the  $\ell_\infty$  norm without its good features. The  $\ell_1$ - and  $\ell_\infty$ -norm trust regions both have sharp corners in which to trap unwary iterates. However, the  $\ell_\infty$ -norm trust region is easy to use, as a point may be checked component by component to see if it lies in the region. The  $\ell_2$  norm is not as easy to apply, but has a strong theoretical advantage in that there are provably efficient methods for minimizing  $q$  within an  $\ell_2$  trust region.<sup>81</sup>

In this chapter, we shall predominantly consider the  $\ell_2$ -norm trust-region subproblem. We shall turn to the  $\ell_\infty$ -norm trust-region subproblem right at the end of the chapter. We leave other norms to the reader's imagination.

## Notes and References for Section 7.1

The relative complexity of solving the subproblems in different norms is summarized by Vavasis (1992b). For a more general discussion of polynomial and nonpolynomial algorithms, and its implications, see Garey and Johnson (1979) and Papadimitriou and Steiglitz (1982).

---

<sup>81</sup>To be precise, minimizing  $q$  within an  $\ell_2$  trust region requires at most a polynomial (in  $n$ ) number of operations, while the equivalent problems in the  $\ell_1$  and  $\ell_\infty$  norms are part of the NP hard class of problems, for which it is thought that there is no polynomial algorithm.

Very roughly, a *polynomial* algorithm for solving a problem is one in which the worst-case running time can be shown to be a polynomial function of the size of the data. For instance, there are polynomial algorithms for solving systems of linear equations and for solving linear and convex quadratic programming problems. At the other extreme, some algorithms have a worst-case running time that is an exponential (or worse) function of the data; the simplex method for linear programming with all common pivoting rules is exponential. Between these two extremes lies a large class of problems that are called NP. The class NP belongs to the set of what are called decision problems. *Decision* problems are problems for which an answer “yes” or “no” must be generated. (For instance, given any  $A$  and  $b$ , we might be asked whether there is an  $x$  for which  $Ax \leq b$ .) The class NP comprises precisely those decision problems for which a “yes” answer for a particular problem instance may be checked in polynomial time. (For example, given that we are told that the particular system  $Ax \leq b$  has a solution, we need to verify this in a polynomial number of steps.) While the class P of all decision problems that have polynomial solution algorithms lies in NP, it is not known if the two classes are the same. A subset of NP, the set of so-called *NP complete* problems, are those decision problems for which any other problem in NP may be transformed to the given problem by a polynomial-time algorithm. The class NP complete contains such famous “hard” problems as integer linear programming and the travelling salesman problem. Significantly, unless  $P = NP$ , no problem in NP complete is solvable in polynomial time. Not every problem lies in NP, but many of them have the property that all NP problems may be reduced to the given problem by a polynomial-time algorithm. These problems are known as *NP hard* problems. The example of most interest to us is the nonconvex quadratic programming problem. Of course, all NP complete problems are NP hard, and in fact there is a decision variant of the nonconvex quadratic programming problem that lies in NP complete.

We should issue a few words of caution. Just because an algorithm is polynomial, it does not mean that it is an effective method in practice (e.g., the ellipsoidal method for linear programming is polynomial but extremely poor in almost every case), nor does it mean that an exponential algorithm is bad except in the worst case (the simplex method for linear programming has an excellent average-case behaviour; see Smale, 1983). Thus, although complexity theory plays an important role in the design of algorithms, it should not be regarded as the only goal. Just because a problem is NP hard, this does not mean that people do not want to solve it, but merely implies that our algorithm will typically rely on heuristics and may behave badly in the (fortunately infrequent) worst case.

## 7.2 Characterization of the $\ell_2$ -Norm Model Minimizer

To find the model minimizer of the  $\ell_2$ -norm trust-region problem, we seek the solution,  $s^M$ , of the minimization problem

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \\ & \text{subject to} && \|s\|_2 \leq \Delta. \end{aligned} \tag{7.2.1}$$

For simplicity, we have dropped the subscripts  $k$  and discarded the constant term  $f(x)$  from the model (7.1.1), as this term has no effect on the value<sup>82</sup> of  $s^M$ . Of course, once we have found  $s^M$ ,  $x^M$  is simply  $x + s^M$ .

A number of things are immediately apparent about (7.2.1). The solution we are seeking lies either interior to the trust region, that is  $\|s\|_2 < \Delta$ , or on the boundary,  $\|s\|_2 = \Delta$ . If the solution is interior, the trust-region bound may as well not have been there, and therefore  $s^M$  is the unconstrained minimizer of  $q(s)$ . But this can only happen if  $q(s)$  is convex, that is, if the Hessian  $H$  is positive semidefinite—we could be more precise here, but the details will shortly emerge. At once we see that a minimizer of a convex model problem may have a different character from that of a nonconvex one. In the nonconvex case, a solution must lie on the boundary of the trust region, while in the convex case a solution may or may not do so.

This in turn suggests an algorithm to solve the trust-region problem. Firstly, find the unconstrained minimizer of the model. If the model is unbounded from below, or if the unconstrained minimizer lies outside the trust region, the model minimizer must occur on the boundary and thus can be found as the global minimizer of  $q(s)$  subject to  $\|s\|_2 = \Delta$ . What can we say about a solution to this constrained optimization problem?

**Theorem 7.2.1** Any global minimizer of  $q(s)$  subject to  $\|s\|_2 = \Delta$  satisfies the equation

$$H(\lambda^M)s^M = -g, \quad (7.2.2)$$

where  $H(\lambda^M) \stackrel{\text{def}}{=} H + \lambda^M I$  is positive semidefinite. If  $H(\lambda^M)$  is positive definite,  $s^M$  is unique.

**Proof.** We first rewrite the constraint  $\|s\|_2 = \Delta$  as  $c(s) \equiv \frac{1}{2}\|s\|_2^2 - \frac{1}{2}\Delta^2 = 0$ . Now, we introduce a Lagrange multiplier  $\lambda^M$  for the constraint and use the first-order optimality conditions (Section 3.2.2). This gives

$$\nabla_s q(s^M) + \lambda^M \nabla_s c(s^M) \equiv Hs^M + g + \lambda^M s^M = 0, \quad (7.2.3)$$

which is (7.2.2).

Next, we consider feasible perturbations of  $q(s)$  about  $x + s^M$ . Suppose  $s^F$  is a feasible point, that is, that  $\|s^F\|_2 = \Delta$ . As  $m$  is quadratic, we obtain

$$q(s^F) = q(s^M) + \langle s^F - s^M, \nabla_s q(s^M) \rangle + \frac{1}{2} \langle s^F - s^M, \nabla_{ss} q(s^M)(s^F - s^M) \rangle. \quad (7.2.4)$$

Moreover, (7.2.3) gives that  $\nabla_s q(s^M) = -\lambda^M s^M$ . This and the restriction that  $s^F$

---

<sup>82</sup>Alternatively, we may view  $q(s)$  as the predicted reduction  $m(x + s) - m(x)$ .

and  $s^M$  are feasible (i.e.,  $\|s^F\|_2 = \Delta = \|s^M\|_2$ ) then imply that

$$\begin{aligned}\langle s^F - s^M, \nabla_s q(s^M) \rangle &= \langle s^M - s^F, s^M \rangle \lambda^M = (\Delta^2 - \langle s^F, s^M \rangle) \lambda^M \\ &= \left[ \frac{1}{2} (\langle s^F, s^F \rangle + \langle s^M, s^M \rangle) - \langle s^F, s^M \rangle \right] \lambda^M \\ &= \frac{1}{2} \langle s^F - s^M, s^F - s^M \rangle \lambda^M.\end{aligned}\quad (7.2.5)$$

Combining (7.2.4) and (7.2.5) with the identity  $\nabla_{ss} q(s^M) = H$  gives

$$\begin{aligned}q(s^F) &= q(s^M) + \frac{1}{2} \langle s^F - s^M, (H + \lambda^M I)(s^F - s^M) \rangle \\ &= q(s^M) + \frac{1}{2} \langle s^F - s^M, H(\lambda^M)(s^F - s^M) \rangle.\end{aligned}\quad (7.2.6)$$

The second-order necessary optimality conditions for a minimizer require that the Hessian

$$\nabla_{ss} q(s) + \lambda^M \nabla_{ss} c(s) \equiv H(\lambda^M)$$

be positive semidefinite on the null-space of  $\nabla_s c(s) \equiv s^M$ . That is,  $\langle z, H(\lambda^M)z \rangle \geq 0$  for all  $z$  for which  $\langle z, s^M \rangle = 0$ . It remains to consider vectors  $v$  for which  $\langle v, s^M \rangle \neq 0$ . To this end, define the line  $s^M + \alpha v$  as a function of the scalar  $\alpha$ . Because we are considering  $v$  for which  $\langle v, s^M \rangle \neq 0$ , this line intersects the constraint  $\|s\|_2 = \Delta$  for two values of  $\alpha$ ,  $\alpha = 0$  at which  $s = s^M$ , and  $\alpha = \alpha^F \neq 0$  at which  $s = s^F$  (see Figure 7.2.1). So  $s^F - s^M = \alpha^F v$ , and therefore, using (7.2.6) we have that

$$q(s^F) = q(s^M) + \frac{1}{2} (\alpha^F)^2 \langle v, H(\lambda^M)v \rangle.$$

Finally, as we are assuming that  $s^M$  is a global minimizer, we must have that  $q(s^F) \geq q(s^M)$ , and thus that  $\langle v, H(\lambda^M)v \rangle \geq 0$ . In summary, we have shown that  $\langle v, H(\lambda^M)v \rangle \geq 0$  for any vector, which is the same as saying that  $H(\lambda^M)$  must be positive semidefinite.

Conversely, if  $H(\lambda^M)$  is positive definite,  $\langle s^F - s^M, H(\lambda^M)(s^F - s^M) \rangle > 0$  for any  $s^F \neq s^M$ , and therefore (7.2.6) shows that  $q(s^F) > q(s^M)$  whenever  $s^F$  is feasible. Thus  $s^M$  is the unique global minimizer.  $\square$

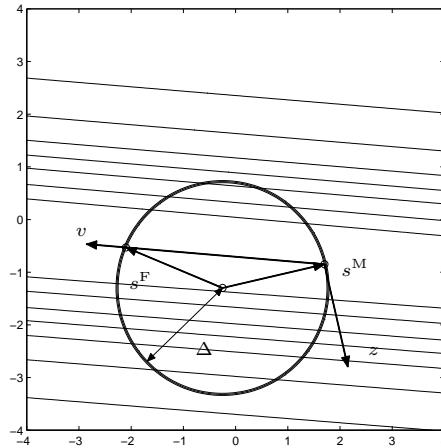


Figure 7.2.1:  $v$  intersects the trust-region boundary in two places whenever  $\langle v, s^M \rangle \neq 0$ .

If we combine Theorem 7.2.1 with the preceding discussion, we have the following immediate result.

**Corollary 7.2.2** Any global minimizer of  $q(s)$  subject to  $\|s\|_2 \leq \Delta$  satisfies the equation

$$H(\lambda^M)s^M = -g, \quad (7.2.7)$$

where  $H(\lambda^M)$  is positive semidefinite,  $\lambda^M \geq 0$ , and  $\lambda^M(\|s^M\|_2 - \Delta) = 0$ . If  $H(\lambda^M)$  is positive definite,  $s^M$  is unique.

**Proof.** If a global solution lies on the trust-region boundary, Theorem 7.2.1 shows that (7.2.7) is satisfied and that  $H(\lambda^M)$  is positive semidefinite. Furthermore,  $\lambda^M(\|s^M\|_2 - \Delta) = 0$  is obviously true, while the Lagrange multiplier condition  $\lambda^M \geq 0$  is a necessary condition for the trust-region constraint to be active (see Section 3.2.2).

If a global solution lies interior to the trust-region bound, it is necessary that the solution satisfy  $\nabla_s q(s^M) \equiv Hs^M + g = 0$ , with  $H$  positive semidefinite. These conditions are precisely those stated above when  $\lambda^M$  has the value zero. Finally, if  $H(\lambda^M)$  is positive definite, then Theorem 7.2.1 shows that at most one global minimizer can occur on the trust-region boundary. If a global minimizer lies in the interior, the condition on  $H(\lambda^M)$  with  $\lambda^M = 0$  ensures that  $H$  must itself be positive definite, and this minimizer is therefore unique. Thus, in all cases,  $s^M$  is unique when  $H(\lambda^M)$  is positive definite.  $\square$

Notice that if there is an interior solution,  $s^M$ , and  $H$  is positive semidefinite but singular, then there is a pair of solutions on the trust-region boundary, each with a zero Lagrange multiplier. For, if  $H$  is singular, there must be a nonzero vector  $v$  such that  $Hv = 0$ . As  $Hs^M + g = 0$ , it follows that  $H(s^M + \alpha v) + g = 0$  for all  $\alpha$ . The two roots of the quadratic equation  $\|s^M + \alpha v\|_2 = \Delta$  provide the required solutions on the trust-region boundary.

We stress that Theorem 7.2.1 and its corollary are more than just an immediate application of (local) first-order optimality conditions. In particular,  $H(\lambda)$  is seen to be positive definite in more than simply the tangent plane to the trust-region boundary. In fact, it is quite rare to be able to find a characterization of the global solution of a nonconvex optimization problem, and we are indeed fortunate in this case.

It is also possible to interpret Theorem 7.2.1 and its corollary in a different way, namely, that the parameter  $\lambda^M$  is used to “regularize” or “modify” the model so that the modified model is convex and so that its minimizer lies on or within the trust-region boundary. When the modification makes the model strictly convex, the solution is unique. Otherwise, the modified model may have more than one solution. We illustrate this in Figures 7.2.2 and 7.2.3.

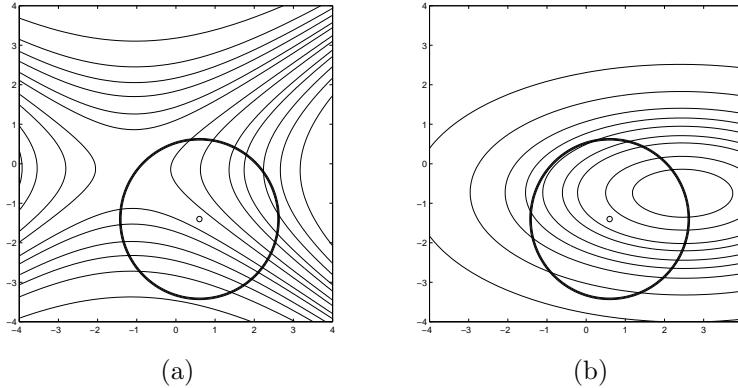


Figure 7.2.2: (a) The contours of a model and the trust-region boundary, and (b) the contours of the (strictly convex) modified model.

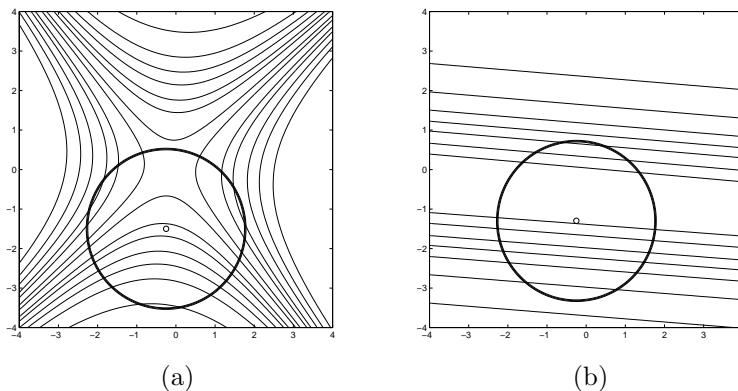


Figure 7.2.3: (a) The contours of a model with two solutions and the trust-region boundary, and (b) the contours of the singular (convex) modified model.

## Notes and References for Section 7.2

The characterization of the solution contained in Corollary 7.2.2 was obtained by Goldfeldt, Quandt, and Trotter (1966), which was a generalization of a result due to Marquardt (1963) for convex problems (see also Morrison, 1960). The complete result was given, independently, by Gay (1981) and Sorensen (1982a). Properties of the path—the so-called *hook step*—swept out by the model minimizer as a function of the radius were reviewed by Andrews and Vicente (1999).

There are a number of other interesting properties concerning critical points of the problem (7.2.1). Martínez (1994) has shown that there is at most one local minimizer aside from the global one, while Lucidi, Palagi, and Roma (1994, 1998) have demonstrated that there are at most  $\min[2n_+ + 1, 2n_- + 2]$  critical points with distinct values of  $q$ , where  $n_-$  is the number of distinct negative eigenvalues of  $H$ . This latter result has some merit, as they also show that it is easy to escape from nonoptimal critical points, and thus any method that hunts for critical rather than minimizing points can be redirected from the former toward the latter. See also Lyle and Szularz (1994) and Pham Dinh and Le Thi (1995).

Ben-Tal and Teboulle (1996), Stern and Wolkowicz (1995), and Flippo and Jansen (1996) offered an explanation as to why it is possible to find the global solution to the nonconvex optimization problem (7.2.1), based on the observation that there is an associated dual optimization problem without a duality gap. They concluded that (7.2.1) really sits “between” convex and nonconvex problems. Vavasis and Zippel (1990) were the first to show formally that there are polynomial-time algorithms (for instance, that due to Ye, 1989) for the problem. See also Yuan (1998b) for a survey of the subproblem solution from the linear algebra point of view.

### 7.3 Finding the $\ell_2$ -Norm Model Minimizer

Having characterized a solution to (7.2.1) in Corollary 7.2.2, we now turn to a means of finding it. The algorithm we shall develop over the next few pages is relatively complicated in its detail, although it is based upon a number of well-known and simple ideas.

The aim is to use the characterization implied in Corollary 7.2.2 to derive a nonlinear equation in a single unknown, the scalar  $\lambda$ , and to solve this equation using Newton’s method (see Section 7.3.1). The most obvious equation, in which  $s(\lambda) = -H(\lambda)^{-1}g$  is substituted into  $\lambda(\|s(\lambda)\|_2 - \Delta) = 0$ , turns out to have severe disadvantages, but a simple alternative, the secular equation, is most effective (Section 7.3.3). As Newton’s method alone offers no guarantee of convergence, the method is safeguarded by finding guaranteed lower and upper bounds on  $\lambda$  (Section 7.3.4) and ensuring that in the worst case these bounds ultimately coincide, thereby trapping the sought value  $\lambda^M$  (Section 7.3.5). A further difficulty arises as  $H(\lambda^M)$  may be singular, which means that  $s^M$  may not be determined directly from (7.2.7) and may ultimately require that an appropriate eigenvector of  $H$  is approximated (Section 7.3.1.3). All of these ingredients, plus rules for finding initial estimates of  $\lambda^M$  (Section 7.3.7) and its safeguards (Section 7.3.8), form the basis of the algorithm we ultimately propose in Section 7.3.9. Finally, rather than finding an exact model minimizer, it is often acceptable to stop with a good approximation. In Section 7.3.10, we therefore develop suitable rules that allow for early termination.

Having set the scene, we now consider our method in full detail.

#### 7.3.1 Finding the $\ell_2$ -Norm Model Minimizer

Corollary 7.2.2 tells us that we should be looking for solutions to (7.2.7), and implicitly tells us what value of  $\lambda$  we need. Suppose that  $H$  has an eigendecomposition

$$H = U^T \Lambda U, \quad (7.3.1)$$

where  $\Lambda$  is a diagonal matrix of eigenvalues  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  and  $U$  is an orthonormal matrix of associated eigenvectors. Then

$$H(\lambda) = U^T (\Lambda + \lambda I) U.$$

We deduce immediately from Corollary 7.2.2 that the value of  $\lambda$  we seek must satisfy  $\lambda^M \geq -\lambda_1$  (as only then is  $H(\lambda)$  positive semidefinite), and if  $\lambda^M > -\lambda_1$ , the model minimizer is unique (as this ensures that  $H(\lambda)$  is positive definite).

Suppose that  $\lambda > -\lambda_1$ . Then  $H(\lambda)$  is positive definite, and thus (7.2.7) has a unique solution,

$$s(\lambda) = -H(\lambda)^{-1}g = -U^T(\Lambda + \lambda I)^{-1}Ug.$$

But, of course, the solution we are looking for depends upon the *nonlinear* inequality

$$\|s(\lambda)\|_2 \leq \Delta.$$

To say more, we need to examine  $\|s(\lambda)\|_2$  in detail. For convenience, define  $\psi(\lambda) \stackrel{\text{def}}{=} \|s(\lambda)\|_2^2$ . We have that

$$\psi(\lambda) = \|U^T(\Lambda + \lambda I)^{-1}Ug\|_2^2 = \|(\Lambda + \lambda I)^{-1}Ug\|_2^2 = \sum_{i=1}^n \frac{\gamma_i^2}{(\lambda_i + \lambda)^2}, \quad (7.3.2)$$

where  $\gamma_i$  is  $[Ug]_i$ , the  $i$ th component of  $Ug$ .

### 7.3.1.1 A Convex Example

We consider a simple, but typical, convex example. Suppose that our model problem is defined by the following data:

$$g = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.$$

We plot the function  $\psi(\lambda)$  in Figure 7.3.1. Notice how the function is always positive, how it vanishes as  $\lambda$  approaches infinity, and how it has a pole at the negatives of each of the eigenvalues, 1, 2, 3, and 4, of  $H$ . If we examine the general form (7.3.2), we can see why this is so. Each term  $\gamma_i^2/(\lambda_i + \lambda)^2$ , and therefore the sum that makes up  $\psi(\lambda)$ , is positive. Furthermore, each term individually vanishes as  $\lambda$  approaches infinity. Finally, one or more of the denominators  $(\lambda_i + \lambda)^2$  vanishes when  $\lambda = -\lambda_i$ , and thus there will be a pole at these values so long as<sup>83</sup>  $\gamma_i \neq 0$ . We thus see that  $\psi$  takes any positive value we like for a unique value of  $\lambda$  between its largest pole and infinity. For our example, the largest pole is  $-1$ , which is the negative of the smallest eigenvalue of  $H$ .

Now let us interpret Corollary 7.2.2 in terms of Figure 7.3.1. In view of the corollary, we are only interested in positive  $\lambda$ . If  $\Delta$  is large, we see that  $\psi(\lambda) < \Delta^2$  for all  $\lambda \geq 0$ . As we reduce  $\Delta$ , we reach a critical value for which, for the first time,  $\psi(\lambda) = \Delta^2$ ; this occurs for  $\Delta^2 \approx 1.5$ . For  $\Delta$  smaller than this value, we must increase  $\lambda$  to maintain

---

<sup>83</sup>This exception is an indication of difficulties to come.

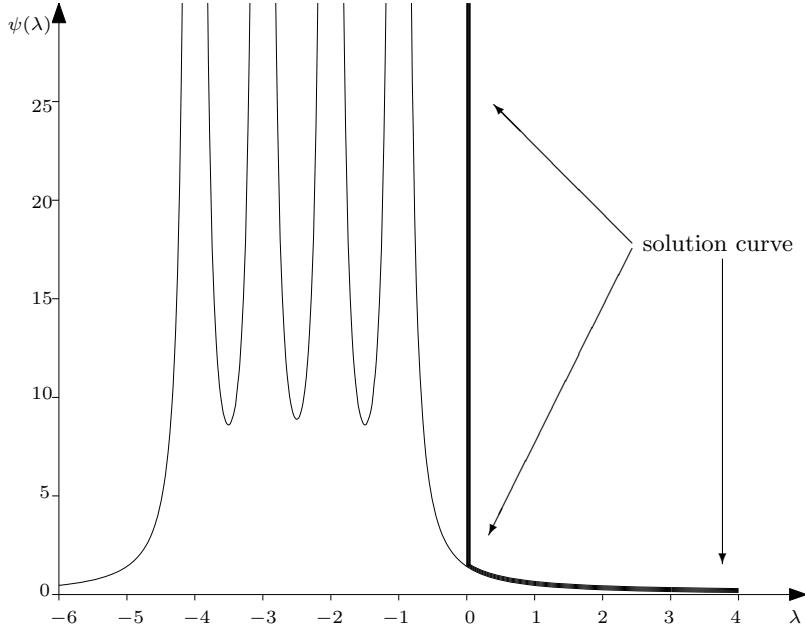


Figure 7.3.1: A plot of  $\psi(\lambda)$  as  $\lambda$  varies from  $-6$  to  $4$ . Note the poles at the negatives of the eigenvalues of  $H$ . The heavy curve plots  $\lambda$  against  $\Delta$ ; the vertical component corresponds to interior solutions, while the remaining segment indicates boundary solutions.

$\psi(\lambda) = \Delta^2$ , and each  $\Delta$  smaller than the critical value corresponds to a unique  $\lambda$ . Geometrically, as  $H$  is positive definite, the model function has a unique unconstrained minimizer.

If  $\Delta$  is large enough, this minimizer lies within the trust region. As  $\Delta$  shrinks, there is a critical value where the unconstrained minimizer actually lies on the boundary of the trust region. Any  $\Delta$  smaller than this critical value excludes the unconstrained minimizer, and the model minimizer must lie on the trust-region boundary. The model minimizer is then given by (7.2.7) for some strictly positive value of  $\lambda$  which may be determined by solving the nonlinear equation

$$\|s(\lambda)\|_2 - \Delta = 0. \quad (7.3.3)$$

We see the important interaction between  $\lambda$  and  $\Delta$ . Early trust-region methods were formulated in terms of adjusting  $\lambda$ , but it is now more common to adjust  $\Delta$  instead (see Section 1.2). Notice how the curve  $s(\lambda)$  defines a smooth path between the Newton direction (large  $\Delta$  or, alternatively, zero  $\lambda$ ) and the (scaled) steepest-descent direction  $-g$  (small  $\Delta$  or, alternatively, large  $\lambda$ ). We will expand on this observation in Section 7.5.3.

### 7.3.1.2 A Nonconvex Example

Now suppose that we change our example by shifting all of the eigenvalues to the left by 3; that is, we consider the model problem with the following data:

$$g = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We plot the function  $\psi(\lambda)$  for this data in Figure 7.3.2. Again notice how the function is always positive, how it vanishes as  $\lambda$  approaches infinity, and how it has a pole at the negatives of each of the eigenvalues,  $-2$ ,  $-1$ ,  $0$ , and  $1$ , of  $H$ . We thus see that  $\psi$  takes any positive value we like for a unique value of  $\lambda$  between its largest pole and infinity. For this example, the largest pole is  $2$ , which is the negative of the smallest eigenvalue of  $H$ , and thus  $H + \lambda I$  is positive definite for all  $\lambda > 2$ . As  $\Delta$  is positive, we deduce that there is a single value of  $\lambda > 2$  for which  $\psi(\lambda) = \Delta^2$ .

Geometrically,  $H$  is indefinite, so the model function is unbounded from below. Thus the solution must lie on the trust-region boundary. The model minimizer is given by (7.2.7) for some strictly positive value of  $\lambda$  which may be determined by solving the nonlinear equation (7.3.3).

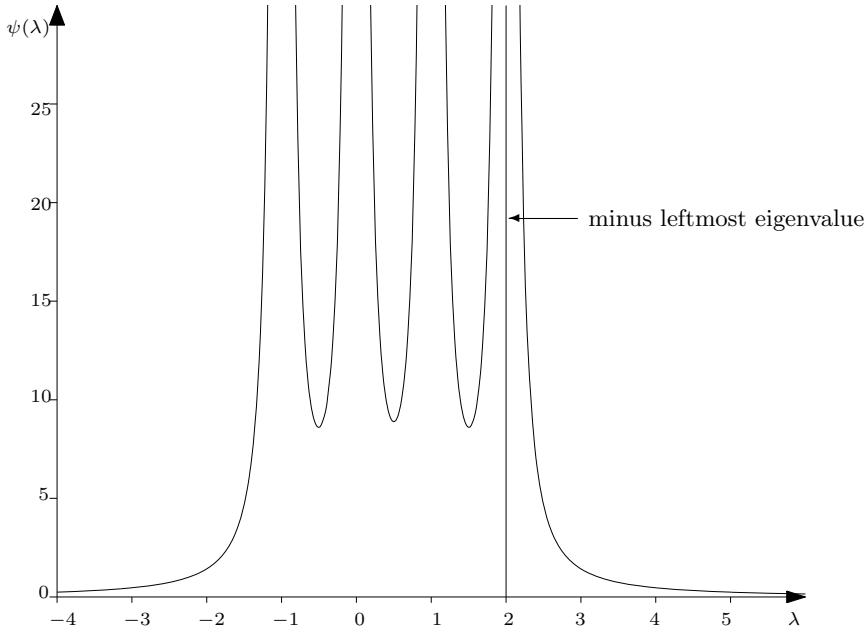


Figure 7.3.2: A plot of  $\psi(\lambda)$  as  $\lambda$  varies from  $-4$  to  $5$ . Again, note the poles at the negatives of the eigenvalues of  $H$ .

### 7.3.1.3 The “Hard” Case

Now suppose that we change the data again so that the first component of  $g$  is 0, not 1. That is, we consider the following data:

$$g = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We plot the function  $\psi(\lambda)$  for this data in Figure 7.3.3. The largest pole here is at  $\lambda = 1$ . So in this case, while there is a single root of (7.3.3) between this pole and infinity, Corollary 7.2.2 requires that  $\lambda \geq 2$ . This seems to suggest difficulties when  $\Delta$  is too large—in our example, when  $\Delta$  is larger than roughly 1.2. So what has gone wrong, and what is the cure?

The problem is easy to explain, and we have already hinted at it in Section 7.3.1.1. For almost any  $g$ , the difficulty will not occur. But for this particular  $g$ , the first component,  $g_1$ , is zero. Because the data is so simple, we have that  $g_i = \gamma_i$  in (7.3.2), and thus  $\gamma_1 = 0$ . Hence the pole that would normally occur in  $\psi(\lambda)$  at minus the most negative eigenvalue has vanished. Notice that this can only happen, in general, if  $\gamma_i = 0$  for all  $i$  for which  $\lambda_i = \lambda_1$ . Or putting this more succinctly, the difficulty can only arise when  $g$  is orthogonal to the space,  $\mathcal{E}_1$ , of eigenvectors corresponding to the

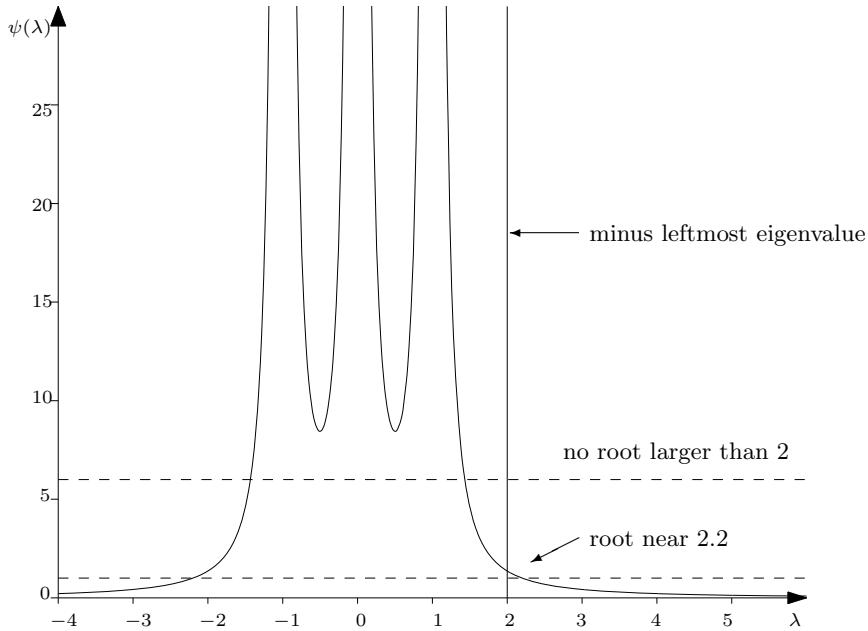


Figure 7.3.3: A plot of  $\psi(\lambda)$  for the modified model as  $\lambda$  varies from  $-4$  to  $5$ . Note that there is no solution to the equation  $\psi(\lambda) = \Delta^2$  with  $\lambda \geq 2$  for  $\Delta$  larger than roughly 1.2.

most negative eigenvalue of  $H$ . This is generally known as the *hard* case by analogy; the *easy* case is when  $g$  has a component in  $\mathcal{E}_1$ .

Now, we consider the cure or, perhaps we should say, the explanation of what is really happening. All is still fine so long as  $\Delta$  is smaller than the critical value<sup>84</sup>  $\Delta_{\text{cri}} = \psi(-\lambda_1)$ . When  $\Delta = \Delta_{\text{cri}}$ , the equation (7.2.7) has a limiting solution  $s_{\text{cri}}$ , where

$$s_{\text{cri}} = \lim_{\lambda \searrow -\lambda_1} s(\lambda) = -U^T(\Lambda - \lambda_1 I)^+ U g,$$

where the superscript “+” denotes the Moore–Penrose generalized inverse; note that it is crucial here that  $g$  be orthogonal to the space  $\mathcal{E}_1$ . However,  $H(-\lambda_1)$  is positive semidefinite and singular, and therefore (7.2.7) has other solutions when  $\lambda = -\lambda_1$ . In particular, if  $u_1$  is an eigenvector corresponding to the eigenvalue  $\lambda_1$ ,  $H(-\lambda_1)u_1 = 0$ , and thus

$$H(-\lambda_1)(s_{\text{cri}} + \alpha u_1) = -g \quad (7.3.4)$$

for any value of the scalar  $\alpha$ . It is these other solutions that come into play when  $\Delta$  exceeds  $\Delta_{\text{cri}}$ . In fact, a model minimizer is given by (7.3.4), where the value of  $\alpha$  is chosen so that

$$\|s_{\text{cri}} + \alpha u_1\|_2 = \Delta; \quad (7.3.5)$$

the model minimizer is not unique, as (7.3.5) has two real roots. The value  $s^M = s_{\text{cri}} + \alpha u_1$  is indeed a model minimizer as it satisfies (7.2.7), with  $\lambda = -\lambda_1$ ,  $H(-\lambda_1)$  is positive semidefinite, and by construction (7.3.5) holds, which together are the necessary optimality conditions required by Corollary 7.2.2.

We thus see that in the hard case, not only do we need to solve a singular system of equations, but we also need to compute a partial eigensolution of  $H$ . We shall return to this shortly. Of course, it seems rather unlikely that the hard case will arise very often in practice, as it requires both that  $H$  be indefinite and that  $g$  be orthogonal to a specific set of eigenvectors. However, it is much more likely that the component of  $g$  in this space will be small. While any component of  $g$  in this space will reintroduce the missing pole, a small component may make it difficult to determine  $\lambda$  accurately. To see this, consider Figure 7.3.4. In this figure, we plot  $\psi(\lambda)$  for different values of the first component,  $\gamma$ , of  $g$ . Notice how the curves move towards the pole at  $\lambda = 2$  as this component gets smaller. This suggests that it will be increasingly tricky to determine the solution to (7.3.3) accurately as the component of  $g$  in  $\mathcal{E}_1$  vanishes.

### 7.3.2 Finding the Root of $\|s(\lambda)\|_2 - \Delta = 0$

How do we find the root of (7.3.3)? The most obvious approach is to apply a “black-box” root finder. Root-finding algorithms vary from simple bisection of a known interval for the required root to sophisticated methods using derivatives of  $\psi$ , of which Newton’s method is the best known. We would prefer to use a sophisticated method, but Newton’s method is based upon a Taylor approximation of  $\psi(\lambda)$ , and its success is

---

<sup>84</sup>The subscript “cri” is short for “critical”.

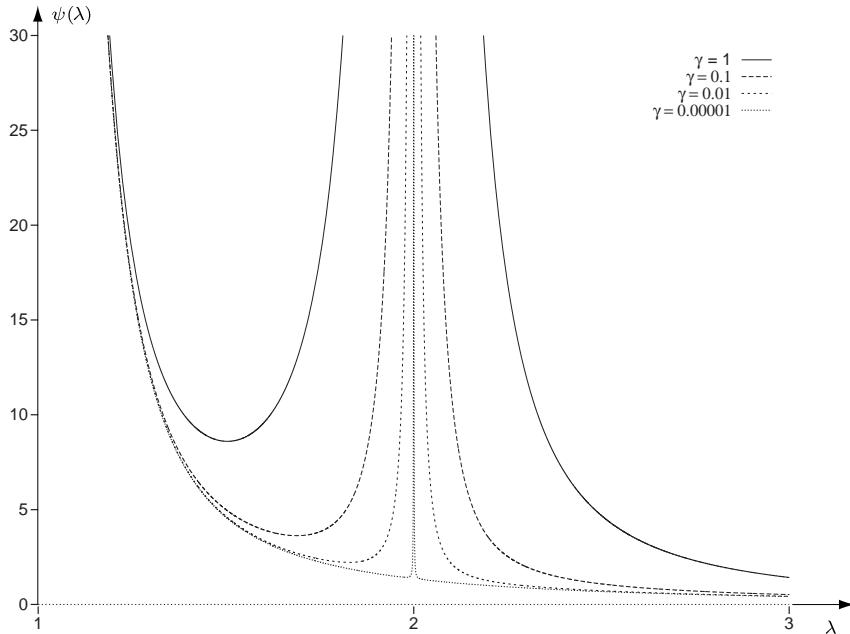


Figure 7.3.4: A plot of  $\psi(\lambda)$  for the modified model as  $\lambda$  varies from 1 to 3 for a variety of values of the first component,  $\gamma$ , of  $g$ .

thus strongly tied to having reasonably behaved derivatives in the region of interest. As there is a pole at  $-\lambda_1$ , large changes in the derivatives will be likely nearby, and this is precisely the region in which we might expect the root to lie. Thus the equation (7.3.3) is particularly unsuited to a direct application of Newton's method. Fortunately, there is an alternative.

### 7.3.3 Newton's Method and the Secular Equation

We have seen that  $\psi(\lambda)$  has poles but no (finite) zeros. This means that the function  $1/\psi(\lambda)$  has zeros but no (finite) poles; the zeros occur at the negatives of the eigenvalues of  $H$ . Therefore, the function  $1/\psi(\lambda)$  is better behaved—it is an analytic function—than  $\psi(\lambda)$ . Thus, rather than solving (7.3.3) directly, it is better to solve the *secular* equation

$$\phi(\lambda) \stackrel{\text{def}}{=} \frac{1}{\|s(\lambda)\|_2} - \frac{1}{\Delta} = 0. \quad (7.3.6)$$

Let us consider, once again, the example from the end of Section 7.3.1.2 in which the first component,  $\gamma$ , of  $g$  is allowed to vary. In Figure 7.3.5, we plot the curves  $1/\|s(\lambda)\|_2$  for a number of values of this component.

We see that, so long as  $g$  has a sizeable component in  $\mathcal{E}_1$ , the curve is effectively linear for  $\lambda$  in the region of interest, that is,  $\lambda > -\lambda_1 = 2$ . This indicates that Newton's method should be extremely efficient when applied to (7.3.6) in this case. However, as

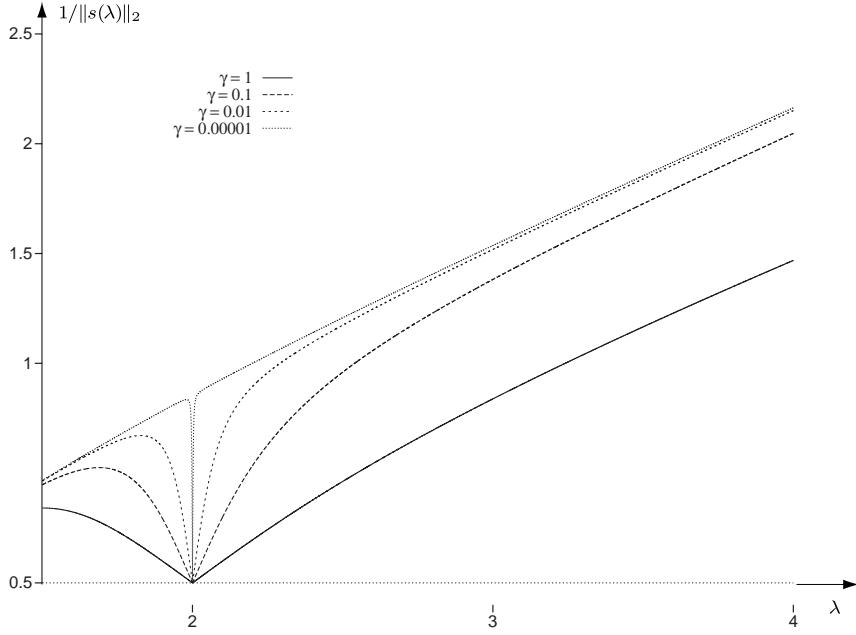


Figure 7.3.5: A plot of  $1/\|s(\lambda)\|_2$  as  $\lambda$  varies from 2 to 4 for a number of different values of the first component,  $\gamma$ , of  $g$ .

the component of  $g$  in  $\mathcal{E}_1$  shrinks, the function becomes more nonlinear, particularly in the interval  $[-\lambda_1, 1/\Delta_{\text{cri}}]$ , which again is an indication of the difficulties associated with the hard case. In general,  $\phi(\lambda)$  has the following properties.

**Lemma 7.3.1** Suppose  $g \neq 0$ . Then the function  $\phi(\lambda)$  is strictly increasing, when  $\lambda > -\lambda_1$ , and concave. Its first two derivatives are

$$\phi'(\lambda) = -\frac{\langle s(\lambda), \nabla_\lambda s(\lambda) \rangle}{\|s(\lambda)\|_2^3} \quad (7.3.7)$$

and

$$\phi''(\lambda) = \frac{3(\langle s(\lambda), \nabla_\lambda s(\lambda) \rangle^2 - \|s(\lambda)\|_2^2 \|\nabla_\lambda s(\lambda)\|_2^2)}{\|s(\lambda)\|_2^5}, \quad (7.3.8)$$

where

$$\nabla_\lambda s(\lambda) = -H(\lambda)^{-1}s(\lambda). \quad (7.3.9)$$

**Proof.** We see from (7.3.2) that  $\psi(\lambda)$  is strictly positive and decreasing for all  $\lambda > -\lambda_1$  provided that  $g \neq 0$ . As

$$\phi(\lambda) = \frac{1}{\sqrt{\psi(\lambda)}} - \frac{1}{\Delta},$$

$\phi(\lambda)$  strictly increases for  $\lambda > -\lambda_1$ . Differentiating  $\phi(\lambda) = \langle s(\lambda), s(\lambda) \rangle^{-\frac{1}{2}} - 1/\Delta$ , we have

$$\phi'(\lambda) = -\langle s(\lambda), s(\lambda) \rangle^{-\frac{3}{2}} \langle s(\lambda), \nabla_\lambda s(\lambda) \rangle,$$

which is (7.3.7), and

$$\phi''(\lambda) = \frac{3\langle s(\lambda), \nabla_\lambda s(\lambda) \rangle^2}{\|s(\lambda)\|_2^5} - \frac{\langle s(\lambda), \nabla_{\lambda\lambda} s(\lambda) \rangle + \|\nabla_\lambda s(\lambda)\|_2^2}{\|s(\lambda)\|_2^3}. \quad (7.3.10)$$

On differentiating the defining equation

$$H(\lambda)s(\lambda) \equiv (H + \lambda I)s(\lambda) = -g,$$

we see that

$$H(\lambda)\nabla_\lambda s(\lambda) + s(\lambda) = 0,$$

which gives (7.3.9), and

$$H(\lambda)\nabla_{\lambda\lambda} s(\lambda) + 2\nabla_\lambda s(\lambda) = 0.$$

These equations give that

$$\begin{aligned} \langle s(\lambda), \nabla_{\lambda\lambda} s(\lambda) \rangle &= -2\langle s(\lambda), H(\lambda)^{-1}\nabla_\lambda s(\lambda) \rangle \\ &= -2\langle H(\lambda)^{-1}s(\lambda), \nabla_\lambda s(\lambda) \rangle \\ &= 2\|\nabla_\lambda s(\lambda)\|_2^2, \end{aligned}$$

which, with (7.3.10), yields (7.3.8). The concavity of  $\phi(\lambda)$  then follows as (7.3.8) and the Cauchy–Schwartz inequality imply that  $\phi''(\lambda) \leq 0$ .  $\square$

Newton's method for finding a root of the scalar equation  $\phi(\lambda) = 0$  replaces the estimate  $\lambda > -\lambda_1$  with the improved estimate  $\lambda^+$  for which

$$\lambda^+ = \lambda - \phi(\lambda)/\phi'(\lambda). \quad (7.3.11)$$

In order to apply Newton's method to (7.3.6), we thus need to evaluate the function  $\phi(\lambda)$  and its first derivative,  $\phi'(\lambda)$ . The value of  $\phi(\lambda)$  may be obtained by solving the equation (7.2.7) to obtain  $s(\lambda)$ , and that of  $\phi'(\lambda)$  is available from (7.3.7) once  $\nabla_\lambda s(\lambda)$  has been found from (7.3.9). Thus both values may be found by solving linear systems involving  $H(\lambda)$ . Fortunately, in the range of interest,  $H(\lambda)$  is positive definite, and thus we may use its Cholesky factors  $H(\lambda) = L(\lambda)L^T(\lambda)$  to solve for (7.3.9) and (7.3.7). The complete Newton algorithm is given on the next page.

Notice that we do not actually need to find  $\nabla_\lambda s(\lambda)$ , but merely the numerator  $\langle s(\lambda), \nabla_\lambda s(\lambda) \rangle = -\langle s(\lambda), H(\lambda)^{-1}s(\lambda) \rangle$  of (7.3.7). The simple relationship

$$\langle s, H(\lambda)^{-1}s \rangle = \langle s, L^{-T}L^{-1}s \rangle = \langle L^{-1}s, L^{-1}s \rangle = \|w\|_2^2$$

explains why we calculate  $w$  in Step 3 of Algorithm 7.3.1. Step 4 follows directly from (7.3.11) and (7.3.7).

**Algorithm 7.3.1:** Newton's method to solve  $\phi(\lambda) = 0$ 

Let  $\lambda > -\lambda_1$  and  $\Delta > 0$  be given.

**Step 1.** Factorize  $H(\lambda) = LL^T$ .

**Step 2.** Solve  $LL^T s = -g$ .

**Step 3.** Solve  $Lw = s$ .

**Step 4.** Replace  $\lambda$  by  $\lambda + \left( \frac{\|s\|_2 - \Delta}{\Delta} \right) \left( \frac{\|s\|_2^2}{\|w\|_2^2} \right)$ .

The Cholesky factorization is useful here for a second, less apparent, reason. In Algorithm 7.3.1, we have presumed that  $\lambda > -\lambda_1$ . But, for a given  $\lambda$ , how do we know? In fact, it is easy to tell, for the Cholesky factors only necessarily exist when  $\lambda > -\lambda_1$ . If  $\lambda < -\lambda_1$ , the factorization does not exist. This becomes apparent when one or more of the squares of the diagonal entries of  $L$  is negative. In the borderline case  $\lambda = -\lambda_1$ , the factorization may or may not exist, but will certainly result in one or more of the diagonals of  $L$  being zero.<sup>85</sup> Thus, to see if  $\lambda > -\lambda_1$ , we merely need to check that the factorization of  $H(\lambda)$  succeeds.<sup>86</sup>

### 7.3.4 Safeguarding Newton's Method

Newton's method by itself is not a reliable method for solving  $\phi(\lambda) = 0$ . The method really needs to be safeguarded to ensure that the iterates it generates do not diverge. However, the concavity of  $\phi$  established in Lemma 7.3.1 has certain useful properties.

**Lemma 7.3.2** Suppose  $\lambda > -\lambda_1$  and  $\phi(\lambda) < 0$ . Then all Newton iterates starting from  $\lambda$  will inherit these properties and converge monotonically towards the required root,  $\lambda^M$ . The convergence is globally Q-linear with factor at least

$$\gamma_\lambda \stackrel{\text{def}}{=} 1 - \frac{\phi'(\lambda^M)}{\phi'(\lambda)} < 1$$

and is ultimately Q-quadratic.

**Proof.** Lemma 7.3.1 implies that  $\phi'(\lambda) > 0$ , and hence it follows from equation (7.3.11) that  $\lambda^+ > \lambda$ . The concavity and differentiability of  $\phi$  (Lemma 7.3.1) together with (7.3.11) then imply that

$$\phi(\lambda^+) \leq \phi(\lambda) + (\lambda^+ - \lambda)\phi'(\lambda) = 0. \quad (7.3.12)$$

<sup>85</sup>The factorization exists for at least one symmetric permutation of  $H$ .

<sup>86</sup>In finite-precision arithmetic a little more care is needed.

A simple Taylor approximation gives that

$$\phi(\lambda) = \phi(\lambda^M) + (\lambda - \lambda^M)\phi'(\lambda^I) = (\lambda - \lambda^M)\phi'(\lambda^I)$$

for some  $\lambda^I \in (\lambda, \lambda^M)$ . Combining this with (7.3.11) gives that

$$\lambda^+ - \lambda^M = (\lambda - \lambda^M) \left( 1 - \frac{\phi'(\lambda^I)}{\phi'(\lambda)} \right).$$

Lemma 7.3.1 also implies that  $\phi'$  is decreasing, and thus, as  $1 - \phi'(\lambda^I)/\phi'(\lambda) \leq \gamma_\lambda$ , the iteration is globally Q-linearly convergent at a rate at least  $\gamma_\lambda$ . The asymptotic Q-quadratic convergence of the Newton iteration follows because the Jacobian of  $\phi$  is nonsingular at the limit point,  $\lambda^M$ ; i.e.,  $\phi'(\lambda^M) \neq 0$ .  $\square$

This lemma is important because it shows that if we can find an iterate between  $-\lambda_1$  and  $\lambda^M$ , convergence is assured. The rate of global Q-linear convergence is interesting. If we consider Figure 7.3.5 again, we see that the gradient of  $\phi$  is close to constant when the  $g$  has a sizeable component in  $\mathcal{E}_1$ . Thus  $\gamma_\lambda$  is small, and the convergence globally fast. On the other hand, when the component of  $g$  in  $\mathcal{E}_1$  is small,  $\gamma_\lambda$  is large, and the convergence is correspondingly slower.

**Lemma 7.3.3** Suppose  $\lambda > -\lambda_1$  and  $\phi(\lambda) > 0$ . Then the next Newton iterate satisfies  $\lambda^+ \leq \lambda^M$  and will additionally satisfy either  $\lambda^+ > -\lambda_1$  and  $\phi(\lambda^+) \leq 0$  or  $\lambda^+ \leq -\lambda_1$ .

**Proof.** The new iterate  $\lambda^+$  is certainly smaller than  $\lambda$  because of (7.3.11) and  $\phi'(\lambda) > 0$ . If  $\lambda^+ > -\lambda_1$ , the concavity of  $\phi$  in this region implies  $\phi(\lambda^+) \leq 0$  because of (7.3.12).  $\square$

At first glance, Lemmas 7.3.2 and 7.3.3 would seem to suggest that all we need to do is find a  $\lambda > -\lambda_1$  for which  $\phi(\lambda) < 0$ , and that if  $\phi(\lambda) > 0$  the Newton step will move us into a more satisfactory region. However, we have to be cautious. In the hard case (Section 7.3.1.3) there may not be any  $\lambda > -\lambda_1$  for which  $\phi(\lambda) < 0$ . Of course, this is not catastrophic, as we have already seen how to find a solution in that case. But it does mean that any algorithm for finding the model minimizer must be able to recognize the hard case. In addition, when  $H$  is convex (Section 7.3.1.1), the required solution may lie in the interior of the trust region (i.e.,  $\phi(\lambda^M) > 0$ ), and  $\lambda^M = 0$  is the required value. Thus in this case it is not the value  $\lambda = -\lambda_1$  that is important, but rather  $\lambda = 0$ .

We shall separate the possible values of  $\lambda$  into three disjoint sets,<sup>87</sup>

$$\begin{aligned}\mathcal{N} &= \{\lambda \mid \lambda \leq \max[0, -\lambda_1]\}, \\ \mathcal{L} &= \{\lambda \mid \max[0, -\lambda_1] < \lambda \leq \lambda^M\}, \quad \text{and} \\ \mathcal{G} &= \{\lambda \mid \lambda > \lambda^M\}.\end{aligned}$$

In addition we let<sup>88</sup>  $\mathcal{F} = \mathcal{L} \cup \mathcal{G}$ . Figure 7.3.6 illustrates these sets.

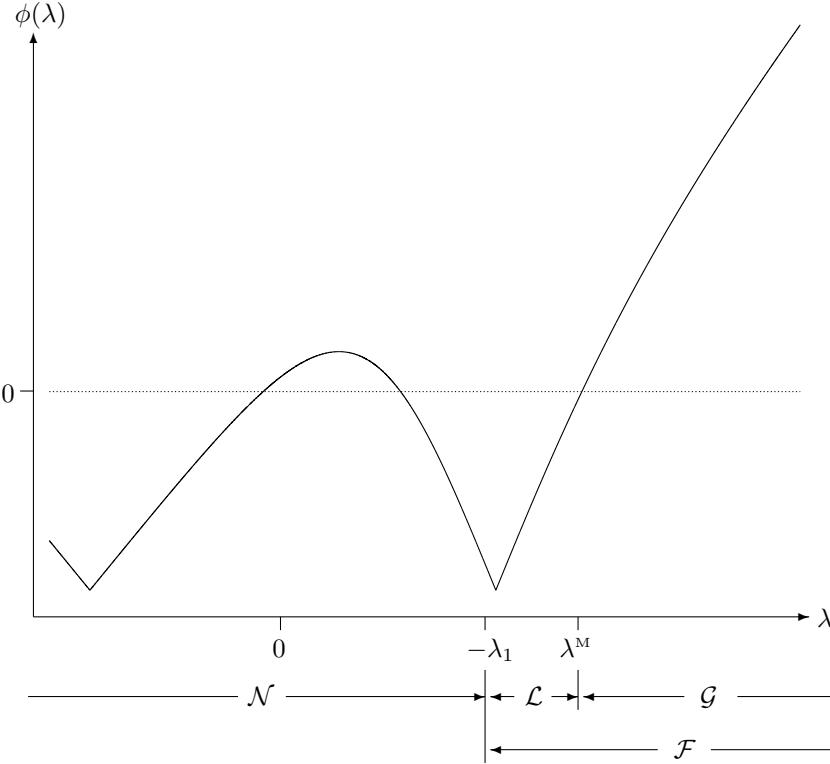


Figure 7.3.6: A plot of  $\phi(\lambda)$  for the problem of minimizing  $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$  within a trust region of radius 0.4, together with the associated sets  $\mathcal{N}$ ,  $\mathcal{L}$ ,  $\mathcal{G}$ , and  $\mathcal{F}$ .

Of course it is easy to establish which of the sets a particular  $\lambda$  falls into, as  $\mathcal{N}$  contains those iterates for which the Cholesky factorization may not exist, while  $\mathcal{L}$  and  $\mathcal{G}$  correspond to the others, with the  $\phi(\lambda)$  being nonpositive and positive, respectively. Lemma 7.3.2 shows that once a Newton iterate  $\lambda$  falls into  $\mathcal{L}$  it stays there, while Lemma 7.3.3 indicates that a Newton iterate from  $\mathcal{G}$  will end up in  $\mathcal{L}$  or  $\mathcal{N}$ . The set  $\mathcal{L}$  is empty in the two important special cases mentioned above, namely, the hard case and the convex interior-solution case, and these are the only cases for which this is so. These cases are illustrated in Figure 7.3.7.

<sup>87</sup>The notation  $\mathcal{N}$  is supposed to indicate “not feasible”, and  $\mathcal{L}$  and  $\mathcal{G}$  to suggest “less than/greater than the model minimizer”.

<sup>88</sup>The notation  $\mathcal{F}$  is intended to suggest “feasible”.

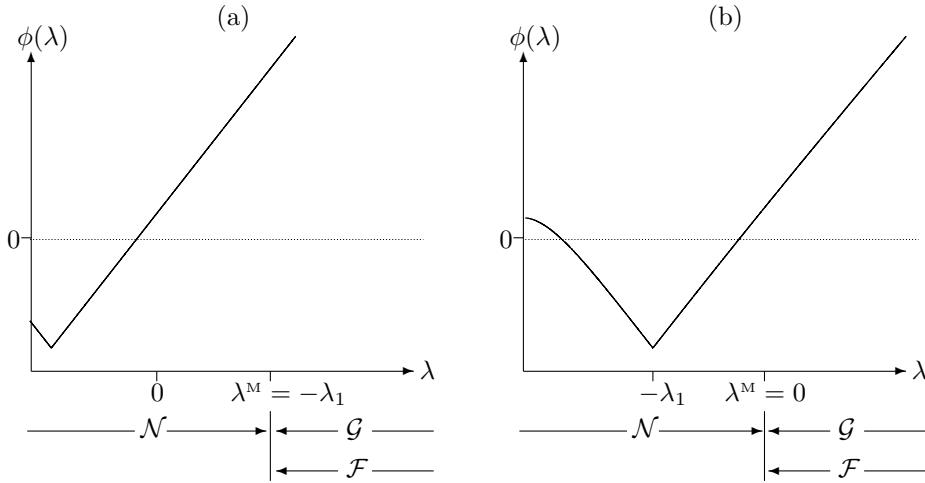


Figure 7.3.7: A plot of  $\phi(\lambda)$  showing the sets  $\mathcal{N}$ ,  $\mathcal{G}$ , and  $\mathcal{F}$  in the two exceptional cases where  $\mathcal{L}$  is empty. (a) The problem of minimizing  $-\frac{1}{4}x_1^2 + \frac{1}{4}x_2^2 + x_2$  within a trust region of radius 0.4 illustrates the hard case. (b) The problem of minimizing  $\frac{3}{4}x_1^2 + \frac{1}{4}x_2^2 + \frac{1}{2}x_1 + x_2$  within a trust region of radius 0.4 illustrates the case where the unconstrained minimizer lies within the trust region.

In order to safeguard the Newton iteration, we construct an interval of uncertainty  $[\lambda^L, \lambda^U]$  in which the solution  $\lambda^M$  is known to occur and in which the current estimate  $\lambda$  is forced to lie. The upper bound  $\lambda^U$  will always be in  $\mathcal{G}$ . If we discover a  $\lambda$  in  $\mathcal{L}$ , all subsequent  $\lambda$  will satisfy  $\lambda^L \leq \lambda \in \mathcal{L}$ . Likewise, if we find a  $\lambda$  in  $\mathcal{G}$ , all subsequent  $\lambda$  will satisfy  $\lambda \leq \lambda^U$ . The aim is to ensure that the interval of uncertainty shrinks at each iteration.

Given the interval of uncertainty, we pick the new  $\lambda$  using Algorithm 7.3.2.

#### Algorithm 7.3.2: Update $\lambda$

- Step 1.** If  $\lambda \in \mathcal{L}$  and  $g \neq 0$ , calculate  $\lambda^+$  using Algorithm 7.3.1 and replace  $\lambda$  with  $\lambda^+$ .
- Step 2.** If  $\lambda \in \mathcal{G}$ , calculate  $\lambda^+$  using Algorithm 7.3.1.
- Step 2a.** If  $\lambda^+ \in \mathcal{L}$ , replace  $\lambda$  with  $\lambda^+$ .
- Step 2b.** If  $\lambda^+ \in \mathcal{N}$  and  $\lambda^+ \leq \lambda^L$ , pick  $\lambda \in [\lambda^L, \lambda^U]$ .
- Step 2c.** If  $\lambda^+ \in \mathcal{N}$  and  $\lambda^+ > \lambda^L$ , reset  $\lambda^L$  to  $\lambda^+$  and pick  $\lambda \in (\lambda^L, \lambda^U)$ .
- Step 3.** If  $\lambda \in \mathcal{N}$  or  $g = 0$ , pick  $\lambda \in (\lambda^L, \lambda^U)$ .

The first alternative in Step 2 of Algorithm 7.3.2 is a realization of Lemma 7.3.2. The new iterate will also lie closer to  $\lambda^M$  in  $\mathcal{L}$ . The second alternative is a direct consequence of Lemma 7.3.3. In Step 2a, the Newton point has landed in the desirable set  $\mathcal{L}$ . In Steps 2b and 2c, the Newton point is not within the feasible set  $\mathcal{F}$ , and we instead prefer to pick a point within the known interval of uncertainty. In Step 2b, we might yet pick  $\lambda^+$ , but we have left open the option of picking a better point if that is possible. In Step 2c, the interval may be contracted, as  $\lambda^+$  provides a new lower bound. In the last alternative, either  $\lambda$  is not within  $\mathcal{F}$  or we have the unlikely possibility that  $g$  is zero. In either case, we aspire to finding a  $\lambda$  in  $\mathcal{F}$ , so we again pick our new point within the interval of uncertainty.

### 7.3.5 Updating the Intervals of Uncertainty

The two safeguarding parameters  $\lambda^L < \lambda^U$  are updated as follows.

**Algorithm 7.3.3: Update  $\lambda^L$  and  $\lambda^U$**

**Step 1.** If  $\lambda \in \mathcal{G}$ , replace  $\lambda^U$  by  $\lambda$ .

**Step 2.** Otherwise, replace  $\lambda^L$  by  $\lambda$ .

The idea here is simple. If  $\lambda \in \mathcal{G}$ , the previous iterate cannot have been a Newton iterate as Lemma 7.3.3 precludes this possibility. Thus the iterate must lie in the previous  $(\lambda^L, \lambda^U)$ , and therefore  $\phi(\lambda) < \phi(\lambda^U)$ . This implies that  $\lambda$  is a smaller upper bound on  $\lambda^M$  than  $\lambda^U$  was. On the other hand, Algorithm 7.3.2 requires that  $\lambda > \lambda^L$ , so that if  $\lambda < -\lambda_1$  or  $\phi(\lambda) \leq 0$ ,  $\lambda$  is a larger lower bound on  $\lambda^M$  than  $\lambda^L$  was.

### 7.3.6 Finding $\lambda$ in the Interval of Uncertainty

There are a number of outstanding issues that we must still address. The first of these is how we pick  $\lambda \in (\lambda^L, \lambda^U)$  in Steps 2b, 2c, and 3 of Algorithm 7.3.2. Consider Steps 2b and 2c first. One possibility is that  $\lambda^L = 0$ ,  $H$  is positive definite, and  $\phi(\lambda) \geq 0$ . When this happens, we simply pick  $\lambda = 0$  and stop; we shall refer to this as *interior* convergence. Otherwise, the value  $\lambda^+$  is not acceptable as anything but a lower bound on  $-\lambda_1$ . We then pick  $\lambda$  so that Algorithm 7.3.3 is sure to contract the interval of uncertainty by a significant amount at the next iteration. The simplest choice is to pick the midpoint

$$\lambda = \frac{1}{2}(\lambda^L + \lambda^U)$$

of the interval as the new point—the interval will half every time we make this choice. However, this disregards the importance of finding a  $\lambda$  in  $\mathcal{L}$  if at all possible, and a better choice is often the geometric mean

$$\lambda = \sqrt{\lambda^L \lambda^U}, \quad (7.3.13)$$

which biases<sup>89</sup> the new point towards  $\lambda^L$  and, it is to be hoped,  $\mathcal{L}$ . However, care must be taken here, as (7.3.13) will lead to a very slow reduction in the interval of uncertainty if  $\lambda^L \ll \lambda^U$ , as then the ratio of the lengths of two successive intervals,

$$\frac{\lambda^U - \sqrt{\lambda^L \lambda^U}}{\lambda^U - \lambda^L} = \frac{\sqrt{\lambda^U}}{\sqrt{\lambda^U} + \sqrt{\lambda^L}},$$

is close to 1. In particular, if  $\lambda^L = 0$ , there is no improvement. Thus it is preferable to replace (7.3.13) by

$$\lambda = \max \left[ \sqrt{\lambda^L \lambda^U}, \lambda^L + \theta(\lambda^U - \lambda^L) \right] \quad (7.3.14)$$

for some small  $\theta \in (0, 1)$ . The update (7.3.14) then guarantees that the ratio of the lengths of two successive intervals is

$$\max \left[ 1 - \theta, \theta, \frac{\sqrt{\lambda^U}}{\sqrt{\lambda^U} + \sqrt{\lambda^L}} \right]. \quad (7.3.15)$$

A value  $\theta = 0.01$  would be appropriate. The alternative  $\lambda = \max \left[ \sqrt{\lambda^L \lambda^U}, \theta \lambda^U \right]$  has also been suggested and provides a similar guaranteed decrease in the length of the interval. The same schemes are entirely appropriate in Step 3, as they provide a mechanism for increasing the lower bound and thus eventually forcing  $\lambda$  into  $\mathcal{F}$ .

### 7.3.7 Finding Good Lower Bounds on $-\lambda_1$

While what we have described so far seems perfectly reasonable, there is a worry that the  $\lambda^L$  may be a poor lower bound on  $-\lambda_1$  and many iterations may be needed to find a  $\lambda$  in  $\mathcal{F}$ . Fortunately, another improvement is possible. First, consider Step 2 of Algorithm 7.3.2. Suppose that while we are picking  $\lambda^+$ , we are able to find a lower bound  $\lambda^B$  on  $-\lambda_1$ . Then, before we attempt to use the lower bound  $\lambda^L$ , we may be able to improve it by replacing it with  $\max[\lambda^L, \lambda^B]$ . One way of finding such a lower bound is to consider the quantity

$$\lambda^B = \lambda - \langle u, H(\lambda)u \rangle$$

for any unit vector<sup>90</sup>  $u$ . As it results from (2.2.5) (p. 19) that  $\langle u, H(\lambda)u \rangle$  is larger than the smallest eigenvalue of  $H(\lambda)$ ,<sup>91</sup> and because this eigenvalue is  $\lambda + \lambda_1$ , we see that  $\lambda^B \leq -\lambda_1$ . Thus  $\lambda^B$  provides just the sort of bound we seek. More importantly, we must try to make  $\langle u, H(\lambda)u \rangle$  as small as possible, as this improves the bound. Indeed, if we find a unit vector  $u$  that minimizes  $\langle u, H(\lambda)u \rangle$ ,  $\lambda^B = -\lambda_1$  and  $u$  is a unit eigenvector of  $H$  corresponding to the eigenvalue  $\lambda_1$ , that is,  $u \in \mathcal{E}_1$ . We thus have an additional bonus. If we are in the hard case, and we use this technique to estimate  $\lambda_1$ , we may obtain an estimate of the eigenvector  $u_1$  required to find the model minimizer.

In Step 2 of Algorithm 7.3.2,  $H(\lambda)$  is positive definite and has a Cholesky factorization  $LL^T$ . There is a cheap and clever way of trying to make  $\langle u, H(\lambda)u \rangle$  small. The

---

<sup>89</sup>The geometric mean is smaller than the arithmetic mean.

<sup>90</sup>Here we are using the  $\ell_2$  norm.

<sup>91</sup>The vector  $u$  is a unit vector.

trick is to note that we are really hoping to find a unit vector in  $\mathcal{E}_1$ , as these eigenvectors minimize the Rayleigh quotient. If we take any unit vector  $v$ , the product  $H(\lambda)^{-1}v$  will tend to emphasize<sup>92</sup> components in  $\mathcal{E}_1$ , and, moreover, the larger  $H(\lambda)^{-1}v$  is, the closer it is to  $\mathcal{E}_1$ . Although it would be too expensive to examine every vector  $v$ , it is possible to encourage large  $H(\lambda)^{-1}v$  for a carefully chosen subset. A method based upon the LINPACK condition estimator (hereafter called the *LINPACK method*) considers  $v$  to be a vector whose components are  $\pm 1$  and aims to make  $H(\lambda)^{-1}v = L^{-T}L^{-1}v$  large by making  $L^{-1}v$  large. This is achieved by ensuring that at each stage of the forward substitution  $Lw = v$ , the sign of  $v$  is chosen to make  $w$  as large as possible. In particular, suppose we have determined the first  $k - 1$  components of  $w$  during the forward substitution. Then the  $k$ th component satisfies

$$l_{kk}w_k = v_k - \sum_{i=1}^{k-1} l_{ki}w_i,$$

and we pick  $v_k$  to be  $\pm 1$  depending on which of

$$\frac{1 - \sum_{i=1}^{k-1} l_{ki}w_i}{l_{kk}} \quad \text{or} \quad \frac{-1 - \sum_{i=1}^{k-1} l_{ki}w_i}{l_{kk}}$$

is larger. Having found  $w$ ,  $u$  is simply  $L^{-T}w/\|L^{-T}w\|_2$ . The vector  $u$  found in this way has the useful properties that

$$u \rightarrow \mathcal{E}_1 \quad \text{and} \quad \langle u, H(\lambda)u \rangle \rightarrow 0 \quad \text{as} \quad \lambda \rightarrow -\lambda_1 \quad (7.3.16)$$

and should be recorded for its potential use in the hard case.

Now we turn to Step 3 of Algorithm 7.3.2. Again we would like to improve on the lower bound  $\lambda^L$  if at all possible. But, in this case,  $H(\lambda)$  is indefinite, and thus its Cholesky factors are of no use in finding a suitable lower bound on  $-\lambda_1$ . Fortunately, there is an alternative in this case that uses a completely different idea. As  $H(\lambda)$  is indefinite, the Cholesky factorization of  $H(\lambda)$  will encounter a nonpositive pivot at the  $k$ th stage of the decomposition for some  $k \leq n$ . It is then possible to add  $\delta = \sum_{j=1}^{k-1} l_{kj}^2 - h_{kk}(\lambda) \geq 0$  to the  $k$ th diagonal of  $H(\lambda)$  so that the leading  $k$  by  $k$  submatrix of

$$H(\lambda) + \delta e_k e_k^T$$

is singular. It is also easy to find a vector  $v$  for which

$$(H(\lambda) + \delta e_k e_k^T) v = 0 \quad (7.3.17)$$

using the Cholesky factors accumulated up to step  $k$ . Setting  $v_j = 0$  for  $j > k$ ,  $v_k = 1$ , and back-solving

$$v_j = -\frac{\sum_{i=j+1}^k l_{ij}v_i}{l_{jj}} \quad \text{for} \quad j = k-1, \dots, 1$$

gives the required vector. We then obtain a lower bound on  $-\lambda_1$  by forming the inner product of (7.3.17) with  $v$ , using the identity  $\langle e_k, v \rangle = v_k = 1$  and recalling that the

---

<sup>92</sup>This is why the power method for finding extreme eigensolutions works.

Rayleigh quotient  $\langle u, H(\lambda)u \rangle / \langle u, u \rangle$  is no smaller than  $\lambda + \lambda_1$ . More precisely, we have that

$$0 = \frac{\langle v, H(\lambda)v \rangle}{\langle v, v \rangle} + \delta \frac{\langle e_k, v \rangle^2}{\langle v, v \rangle} \geq \lambda + \lambda_1 + \frac{\delta}{\|v\|_2^2},$$

which implies the bound

$$\lambda^B \stackrel{\text{def}}{=} \lambda + \frac{\delta}{\|v\|_2^2} \leq -\lambda_1.$$

Thus, in Step 3 of Algorithm 7.3.2, we replace  $\lambda^L$  by  $\lambda^B$  whenever the latter is larger.

### 7.3.8 Initial Values

It remains to provide initial values for  $\lambda$ ,  $\lambda^L$ , and  $\lambda^U$ . Simple bounds on  $\lambda$ , when it is strictly positive, are provided by combining the Rayleigh quotient inequalities

$$(\lambda + \lambda_{\min})^2 \leq \frac{\langle H(\lambda)s(\lambda), H(\lambda)s(\lambda) \rangle}{\langle s(\lambda), s(\lambda) \rangle} = \frac{\langle g, g \rangle}{\langle s(\lambda), s(\lambda) \rangle} \leq (\lambda + \lambda_{\max})^2$$

with the requirement that  $s(\lambda)$  have  $\ell_2$  norm  $\Delta$ . For then

$$\frac{\|g\|_2}{\Delta} - \lambda_{\max} \leq \lambda \leq \frac{\|g\|_2}{\Delta} - \lambda_{\min}. \quad (7.3.18)$$

We can easily replace  $\lambda_{\max}$  and  $\lambda_{\min}$  by suitable easily computable bounds like the Gershgorin bounds (2.2.6) (p. 19), or by upper bounds on the  $\ell_2$  norm like (2.3.4) (p. 23) provided by the Frobenius norm or (2.3.6) (p. 23) given by the  $\ell_\infty$  norm. Suitable initial values of  $\lambda^L$  and  $\lambda^U$  are then

$$\lambda^L = \max \left[ 0, -\min_i [H]_{i,i}, \frac{\|g\|_2}{\Delta} - \min \left[ \max_i \left[ [H]_{i,i} + \sum_{j \neq i} |[H]_{i,j}| \right], \|H\|_F, \|H\|_\infty \right] \right]$$

and

$$\lambda^U = \max \left[ 0, \frac{\|g\|_2}{\Delta} + \min \left[ \max_i \left[ -[H]_{i,i} + \sum_{j \neq i} |[H]_{i,j}| \right], \|H\|_F, \|H\|_\infty \right] \right],$$

where the extra condition on  $\lambda^U$  is to allow for a necessarily interior solution, while those on  $\lambda^L$  ensure that it is positive and that the diagonal entries of  $H(\lambda)$  are nonnegative, which is a necessary condition for  $H(\lambda)$  to be positive semidefinite.

Without any additional information, it is difficult to suggest a good initial value for  $\lambda$  short of requiring that  $\lambda \in [\lambda^L, \lambda^U]$ . If  $\lambda^L$  is zero, it is probably worth picking  $\lambda = 0$  so that any interior solution is found or eliminated quickly. The value  $\lambda = \|g\|_2/\Delta$  has also been suggested.

There is one case where a good initial  $\lambda$  is evident. This is when the model has not changed but the trust-region radius has shrunk. In this case, the terminating value for  $\lambda$  for the larger radius should be chosen for both  $\lambda^L$  and the initial  $\lambda$  for the smaller radius. This is because, in the easy (i.e., not hard) case, this value of  $\lambda$  will lie in  $\mathcal{L}$  for the new value of the radius, and convergence should be rapid from here. In the hard case,  $\lambda^L$  is still a good lower bound.

### 7.3.9 The Complete Algorithm

We are now in a position to draw together all of the preceding discussion into a single algorithm. We suppose that  $\Delta > 0$  and  $0 < \theta < 1$  are given and that we have a bracket  $[\lambda^L, \lambda^U]$  on the model minimizer as well as an estimate  $\lambda \in [\lambda^L, \lambda^U]$  of the minimizer. An iteration of the complete algorithm is outlined in Algorithm 7.3.4. The required model minimizer is given by  $s$ . Notice that we have not yet specified the termination test in Step 4.

**Algorithm 7.3.4: Iteration of the algorithm to find a model minimizer**

**Step 1.** Attempt to factorize  $H(\lambda) = LL^T$  if not already tried.

**Step 1a.** If the factorization succeeds, then  $\lambda \in \mathcal{F}$ . Solve  $LL^T s = -g$ . If  $\|s\|_2 < \Delta$ , then  $\lambda \in \mathcal{G}$ ; check for interior convergence. Otherwise (i.e., the factorization fails),  $\lambda \in \mathcal{L}$ .

Otherwise,  $\lambda \in \mathcal{N}$ .

**Step 2.** If  $\lambda \in \mathcal{G}$ , replace  $\lambda^U$  by  $\lambda$ . Otherwise, replace  $\lambda^L$  by  $\lambda$ .

**Step 3.** If  $\lambda \in \mathcal{F}$ ,

**Step 3a.** Solve  $Lw = s$  and set  $\lambda^+ = \lambda + \left( \frac{\|s\|_2 - \Delta}{\Delta} \right) \left( \frac{\|s\|_2^2}{\|w\|_2^2} \right)$ .

**Step 3b.** If  $\lambda \in \mathcal{G}$ ,

- (i) Use the LINPACK method to find a unit vector  $u$  to make  $\langle u, H(\lambda)u \rangle$  small.
- (ii) Replace  $\lambda^L$  by  $\max[\lambda^L, \lambda - \langle u, H(\lambda)u \rangle]$ .
- (iii) Find the root  $\alpha$  of the equation  $\|s + \alpha u\|_2 = \Delta$  which makes the model  $q(s + \alpha u)$  smallest, and replace  $s$  by  $s + \alpha u$ .

Otherwise (i.e.,  $\lambda \notin \mathcal{F}$ ),

**Step 3c.** Use the partial factorization to find  $\delta$  and  $v$  such that  $(H(\lambda) + \delta e_k e_k^T)v = 0$ .

**Step 3d.** Replace  $\lambda^L$  by  $\max \left[ \lambda^L, \lambda + \frac{\delta}{\|v\|_2^2} \right]$ .

**Step 4.** Check for termination.

**Step 5.** If  $\lambda \in \mathcal{L}$  and  $g \neq 0$ , replace  $\lambda$  with  $\lambda^+$ .

Otherwise, if  $\lambda \in \mathcal{G}$ , attempt to factorize  $H(\lambda^+) = LL^T$ .

**Step 5a.** If the factorization succeeds, then  $\lambda^+ \in \mathcal{L}$ . Replace  $\lambda$  with  $\lambda^+$ .

**Step 5b.** Otherwise,  $\lambda^+ \in \mathcal{N}$ . Replace  $\lambda^L$  by  $\max[\lambda^L, \lambda^+]$ , check  $\lambda^L$  for interior convergence; otherwise replace  $\lambda$  by  $\max \left[ \sqrt{\lambda^L \lambda^U}, \lambda^L + \theta(\lambda^U - \lambda^L) \right]$ .

Otherwise, replace  $\lambda$  by  $\max \left[ \sqrt{\lambda^L \lambda^U}, \lambda^L + \theta(\lambda^U - \lambda^L) \right]$ .

We now consider the convergence of Algorithm 7.3.4. We have the following result.

**Theorem 7.3.4** Suppose that the termination test in Step 4 of Algorithm 7.3.4 is not applied. Then the iterates  $\lambda$  generated by the algorithm converge to  $\lambda^M$ , and the limiting point  $s$  is  $s^M$ . The algorithm converges either in a finite number of steps or, except in the hard case, ultimately at a Q-quadratic rate.

**Proof.** There are three possibilities. Convergence may occur after a finite number of iterations. This can happen if, for instance, the convergence is interior (i.e., the solution lies strictly within the trust region). Secondly, if an iterate enters  $\mathcal{L}$ , Lemma 7.3.2 shows that subsequent iterates remain in this set and converge to  $\lambda^M$  at an asymptotic Q-quadratic rate; the values  $s(\lambda)$  converge to  $s^M$ . Finally, suppose that no iterate of the algorithm falls into  $\mathcal{L}$ . Then, the length of the interval of uncertainty shrinks by at least (7.3.15) every iteration and thus converges to zero. This cannot happen if  $\mathcal{L}$  is nonempty and is only possible in the hard case, since we have excluded finite-interior convergence. However, as  $\lambda^L$  is smaller than  $-\lambda_1$  and  $\lambda^U$  lies in  $\mathcal{F}$ , the interval of uncertainty converges to  $-\lambda$ , at which  $\phi(\lambda) \geq 0$ . As  $\lambda^M$  is not in  $\mathcal{G}$ , an infinite subsequence of the  $\lambda$  must fall into  $\mathcal{G}$ . As  $s$  is only calculated when  $\lambda \in \mathcal{F}$ , Step 3b(iii) will ensure that each value satisfies  $\|s\|_2 = \Delta$ , and thus the limit point will satisfy the conditions of Corollary 7.2.2.  $\square$

### 7.3.10 Termination

Now that we know that Algorithm 7.3.4 will find a model minimizer if run indefinitely, we can confidently expect to be able to terminate the algorithm after a finite number of steps with an acceptable approximation to the required minimizer. We say that  $s$  is an *acceptable* approximation to the model minimizer if

$$q(s) \leq \kappa_{\text{opt}} q(s^M) \quad \text{and} \quad \|s\| \leq \kappa_{\text{tr}} \Delta \quad (7.3.19)$$

for some positive constants<sup>93</sup>  $\kappa_{\text{opt}}$  and  $\kappa_{\text{tr}}$ ; typical values for  $\kappa_{\text{opt}}$  and  $\kappa_{\text{tr}}$  might be  $\frac{1}{2}$  and 1.1, respectively. We shall shortly define termination rules for Algorithm 7.3.4 which guarantee that the calculated  $s$  is an acceptable approximation.

To deal with the easy case, we shall simply stop when we encounter a value  $\lambda \in \mathcal{F}$  for which

$$|\|s(\lambda)\|_2 - \Delta| \leq \kappa_{\text{easy}} \Delta \quad (7.3.20)$$

for some  $\kappa_{\text{easy}} \in (0, 1)$ , or if

$$\|s(\lambda)\|_2 \leq \Delta \quad \text{and} \quad \lambda = 0.$$

This corresponds to solving the “perturbed” problem

---

<sup>93</sup>The subscript “opt” is short for “optimization factor” and “tr” is short for “trust-region factor”.

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad q(s) \\ & \text{subject to} \quad \|s\|_2 \leq \delta \end{aligned}$$

for some  $\delta \in ((1 - \kappa_{\text{easy}})\Delta, (1 + \kappa_{\text{easy}})\Delta)$ . This also implies that any  $s$  satisfying (7.3.20) is an acceptable approximation, as is evident from the following lemma.

**Lemma 7.3.5** Suppose  $\lambda \in \mathcal{F}$ ,  $s(\lambda) = -H(\lambda)^{-1}g$ , and

$$|\|s(\lambda)\|_2 - \Delta| \leq \kappa_{\text{easy}}\Delta \quad (7.3.21)$$

for some  $\kappa_{\text{easy}} \in (0, 1)$ . Then

$$q(s(\lambda)) \leq (1 - \kappa_{\text{easy}})^2 q(s^M). \quad (7.3.22)$$

**Proof.** For any  $v$ , we have the identity

$$\begin{aligned} q(s(\lambda) + v) &= \langle g, s(\lambda) + v \rangle + \frac{1}{2}\langle s(\lambda) + v, H(s(\lambda) + v) \rangle \\ &= \langle g, s(\lambda) + v \rangle + \frac{1}{2}\langle s(\lambda) + v, H(\lambda)(s(\lambda) + v) \rangle - \frac{1}{2}\lambda\|s(\lambda) + v\|_2^2 \\ &= -\langle H(\lambda)s(\lambda), s(\lambda) + v \rangle + \frac{1}{2}\langle s(\lambda) + v, H(\lambda)(s(\lambda) + v) \rangle \\ &\quad - \frac{1}{2}\lambda\|s(\lambda) + v\|_2^2 \\ &= \frac{1}{2}\langle v, H(\lambda)v \rangle - \frac{1}{2}\langle s(\lambda), H(\lambda)s(\lambda) \rangle - \frac{1}{2}\lambda\|s(\lambda) + v\|_2^2. \end{aligned} \quad (7.3.23)$$

Picking  $s^M = s(\lambda) + v^M$  in this equation shows that

$$q(s^M) \geq -\frac{1}{2}(\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\|s(\lambda) + v^M\|_2^2) \geq -\frac{1}{2}(\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2) \quad (7.3.24)$$

as  $\|s^M\|_2 \leq \Delta$ . The requirement (7.3.21) implies that

$$\|s(\lambda)\|_2 \geq (1 - \kappa_{\text{easy}})\Delta,$$

and combining this with (7.3.23) when  $v = 0$  reveals that

$$\begin{aligned} q(s(\lambda)) &= -\frac{1}{2}(\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\|s(\lambda)\|_2^2) \\ &\leq -\frac{1}{2}(\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda(1 - \kappa_{\text{easy}})^2\Delta^2) \\ &\leq -\frac{1}{2}(1 - \kappa_{\text{easy}})^2(\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2). \end{aligned} \quad (7.3.25)$$

The required inequality (7.3.22) is immediate from (7.3.24) and (7.3.25).  $\square$

Thus, provided  $\lambda$  is chosen as in Lemma 7.3.5,  $s(\lambda)$  is an acceptable approximation to the model minimizer so long as  $\kappa_{\text{opt}} \geq (1 - \kappa_{\text{easy}})^2$  and  $\kappa_{\text{tr}} \geq 1 + \kappa_{\text{easy}}$  (using (7.3.21)). Termination in the easy case will occur in a finite number of steps if the test (7.3.21) is employed.

To cope with the hard case, we stop if we encounter a value  $\lambda \in \mathcal{G}$  for which the LINPACK method gives  $u$  and  $\alpha$  such that

$$\alpha^2 \langle u, H(\lambda)u \rangle \leq \kappa_{\text{hard}} (\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2) \quad (7.3.26)$$

for some  $\kappa_{\text{hard}} \in (0, 1)$ . The following result immediately shows why this is useful.

**Lemma 7.3.6** Suppose  $\lambda \in \mathcal{G}$ ,  $s(\lambda) = -H(\lambda)^{-1}g$ , and the vector  $v$  satisfies

$$\langle v, H(\lambda)v \rangle \leq \kappa_{\text{hard}} (\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2) \quad \text{and} \quad \|s(\lambda) + v\|_2 = \Delta \quad (7.3.27)$$

for some  $\kappa_{\text{hard}} \in (0, 1)$ . Then

$$q(s(\lambda) + v) \leq (1 - \kappa_{\text{hard}})q(s^M). \quad (7.3.28)$$

**Proof.** The previously derived (7.3.23) and (7.3.27) together imply that

$$q(s(\lambda) + v) \leq -\frac{1}{2}(1 - \kappa_{\text{hard}}) (\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2). \quad (7.3.29)$$

Moreover, if  $s^M = s(\lambda) + v^M$ , (7.3.24) holds as before. The required inequality (7.3.28) is immediate from (7.3.24) and (7.3.29).  $\square$

Picking  $v = \alpha u$  in Lemma 7.3.6 shows that the stopping rule (7.3.26) yields  $s(\lambda) + \alpha u$  as an acceptable approximation to the model minimizer, so long as  $\kappa_{\text{opt}} \geq 1 - \kappa_{\text{hard}}$  and  $\kappa_{\text{tr}} \geq 1$ . The relationship (7.3.16) reveals that the left-hand side of (7.3.26) converges to zero as  $\lambda$  approaches  $-\lambda_1$ , while the right-hand side is bounded away from zero as  $H(-\lambda_1)$  is positive semidefinite and  $\lim_{\lambda \searrow -\lambda_1} s(\lambda)$  does not lie completely in the space  $\mathcal{E}_1$ . Thus we can be sure that (7.3.26) will be satisfied for some  $\lambda > -\lambda_1$ , and therefore that termination in the hard case will occur in a finite number of steps.

In summary, we use the following termination rules in Step 4 of Algorithm 7.3.4.

#### Algorithm 7.3.5: Termination rules

Let  $\kappa_{\text{easy}}, \kappa_{\text{hard}} \in (0, 1)$ .

**Step 1.** If  $\lambda \in \mathcal{F}$  and

$$|\|s(\lambda)\|_2 - \Delta| \leq \kappa_{\text{easy}}\Delta,$$

stop with  $s = s(\lambda)$ .

**Step 2.** If  $\lambda = 0 \in \mathcal{G}$ , stop with  $s = s(0)$ .

**Step 3.** If  $\lambda \in \mathcal{G}$  and the LINPACK method gives  $u$  and  $\alpha$  such that

$$\alpha^2 \langle u, H(\lambda)u \rangle \leq \kappa_{\text{hard}} (\langle s(\lambda), H(\lambda)s(\lambda) \rangle + \lambda\Delta^2),$$

stop with  $s = s(\lambda) + \alpha u$ .

Given the constants  $\kappa_{\text{easy}}$  and  $\kappa_{\text{hard}}$  in Algorithm 7.3.5, Lemmas 7.3.5 and 7.3.6 show

that the terminating  $s$  satisfies (7.3.19) with

$$\kappa_{\text{opt}} = \max [(1 - \kappa_{\text{easy}})^2, 1 - \kappa_{\text{hard}}] \quad \text{and} \quad \kappa_{\text{tr}} = 1 + \kappa_{\text{easy}}.$$

Values around  $\kappa_{\text{easy}} = 0.1$  and  $\kappa_{\text{hard}} = 0.2$  have been used successfully.

### 7.3.11 Enhancements

The dominant cost of Algorithm 7.3.4 is in forming the Cholesky factorizations in Steps 1 and, perhaps, 5. For dense matrices, the cost is approximately  $n^3/6$  floating-point operations;<sup>94</sup> for sparse matrices the cost is less predictable, but is rarely insignificant. There is an alternative, which is particularly appealing if several iterations of the algorithm are required. Suppose we can decompose

$$H = Q^T \bar{H} Q, \quad (7.3.30)$$

where  $Q$  is orthonormal, and  $\bar{H}$  is cheap to factorize. Then, as

$$H(\lambda) = Q^T (\bar{H} + \lambda I) Q,$$

we may recover the factorization of  $H(\lambda)$  for a range of values of  $\lambda$  merely by factorizing  $\bar{H}(\lambda) = \bar{H} + \lambda I$ . The eigendecomposition (7.3.1) is one example of (7.3.30), but this decomposition is expensive to obtain. A more practical decomposition is when  $\bar{H}$  is banded, the extreme case being when  $\bar{H}$  is tridiagonal. This decomposition may be achieved using Householder transformations (reflections); the reduction to tridiagonal form requires approximately  $2n^3/3$  floating-point operations for dense matrices, that is, roughly four times as much work as the Cholesky factorization. For sparse matrices, Givens' matrices (plane rotations) may be preferred.

Once again consider the model problem

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \\ & \text{subject to} && \|s\|_2 \leq \Delta. \end{aligned} \quad (7.3.31)$$

Suppose that we define

$$\bar{s} = Qs \quad \text{and} \quad \bar{g} = Qg.$$

Then, as  $Q$  is orthonormal,  $s = Q^T \bar{s}$ ,  $\bar{H} = QHQ^T$ , and  $\|s\|_2 = \|\bar{s}\|_2$ . Therefore, we may rewrite (7.3.31) as

$$\begin{aligned} & \underset{\bar{s} \in \mathbb{R}^n}{\text{minimize}} && \bar{q}(\bar{s}) \equiv \langle \bar{g}, \bar{s} \rangle + \frac{1}{2} \langle \bar{s}, \bar{H} \bar{s} \rangle \\ & \text{subject to} && \|\bar{s}\|_2 \leq \Delta \end{aligned} \quad (7.3.32)$$

and recover the solution

$$s^M = Q^T \bar{s}^M. \quad (7.3.33)$$

But this means that we may equivalently apply Algorithm 7.3.4 to (7.3.32) to find  $\bar{s}^M$ , and then recover  $s^M$  from (7.3.33).

---

<sup>94</sup>A floating-point operation is a multiply/add pair.

Significantly,  $Q$  is only needed to form  $\bar{g}$  before applying Algorithm 7.3.4 and to recover  $s^M$  once the required  $\lambda$  has been determined. Thus  $Q$  may be stored as the product of Householder or Givens matrices rather than in assembled form. The dominant cost in Algorithm 7.3.4 is now in factorizing  $\bar{H}(\lambda)$ . Provided  $\bar{H}(\lambda)$  has a narrow bandwidth, this calculation is extremely cheap; for instance, the tridiagonal case requires approximately  $2n$  flops.

One case where it is particularly appealing to use (7.3.30) is when the model Hessian is obtained from a secant updating formula. In this case, one model Hessian is usually obtained from its predecessor via a rank 1 or rank 2 update (see Section 8.4.1.2). Rather than recompute the factorization of the Hessian every time the model changes, it is usual to update an existing factorization to reflect such a low-rank change. This usually cuts the linear algebra costs from  $O(n^3)$  to  $O(n^2)$  floating-point operations. For linesearch algorithms, the Cholesky factorization is particularly appealing since normally the only time the Hessian is required is when solving a single system of equations. For the methods we are considering here, we may require a sequence of factorizations of  $H(\lambda)$  for differing  $\lambda$ , and the cost of each will rise to  $O(n^3)$  floating-point operations when  $\lambda \neq 0$  unless we are extremely careful. In particular, if  $H$  is of the form (7.3.30) and changes by a low-rank matrix, it might be hoped that the factorization (7.3.30) may also be updated in  $O(n^2)$  floating-point operations so long as  $\bar{H}$  has low bandwidth. Unfortunately, this appears not to be the case, although methods have been found where the work drops to  $O(n^{7/3})$  operations if  $Q$  is explicitly obtained and  $O(n^{11/5})$  operations if it is stored as a product of plane rotations. As we have already stressed, once we have a decomposition of the form (7.3.30), calculating factorizations of  $H(\lambda)$  is extremely cheap.

Theorem 7.3.4 indicates that Algorithm 7.3.4 is less likely to be effective in the hard case, as the algorithm is then not Q-quadratically convergent. Indeed, the required value  $\lambda = -\lambda_1$  has less to do with  $s(\lambda)$  than with  $H(\lambda)$  itself. In the methods we have considered so far, we have deliberately avoided computing  $\lambda_1$  and its eigenvector(s). This was reasonable as the computation of such an eigensolution is normally significantly more expensive than the solution of the linear systems that characterize Algorithm 7.3.4. However, for certain classes of matrices, particularly band matrices, computing an eigensolution may in fact be a reasonable expense, particularly if it avoids the disadvantages of the hard case.

Let us suppose that computing the leftmost eigenvalue and, if required, an associated eigenvalue of  $H$  is inexpensive. Then, if this leftmost eigenvalue is strictly positive, the value  $\lambda = 0$  either reveals an interior solution or, because of Lemma 7.3.2, provides a starting point for the globally and quadratically Newton iteration given in Algorithm 7.3.1. If, on the other hand,  $\lambda_1 \leq 0$ , we may be in the hard case. To verify whether we are in the hard case or not, we recall Section 7.3.1.3 and solve the system  $H(-\lambda_1^+)s(-\lambda_1^+) = -g$  for some  $\lambda_1^+$  barely smaller than  $\lambda_1$ . If the resulting  $\|s(-\lambda_1^+)\|_2 \leq \Delta$ , we are in the hard case<sup>95</sup> and need to compute an eigenvector  $u_1$ ,

---

<sup>95</sup>Strictly speaking, this is only true if we let  $\lambda_1^+$  converge from below to  $\lambda_1$ , but so long as  $|\lambda_1^+ - \lambda_1|$  is small, the solution will be acceptable.

corresponding to  $\lambda_1$ , to complete the solution according to (7.3.5). If, on the other hand,  $\|s(-\lambda_1^+)\|_2 > \Delta$ , the value  $\lambda = -\lambda_1^+$  lies in  $\mathcal{L}$  and again provides a starting point for the globally and quadratically Newton iteration, Algorithm 7.3.1. Thus such a scheme either terminates in a finite number of steps (specifically one solution of a linear system and one eigensolution) or is globally and Q-quadratically convergent. We summarize the complete method in Algorithm 7.3.6.

**Algorithm 7.3.6: Find model minimizer when eigensolution is cheap**

Let  $\kappa_{\text{easy}} \in (0, 1)$ .

**Step 1.** If  $H$  is positive definite, set  $\lambda = 0$ . Otherwise, compute its leftmost eigenvalue  $\lambda_1$  and set  $\lambda = -\lambda_1^+$ .

**Step 2.** Factorize  $H(\lambda) = LL^T$  and solve  $LL^T s = -g$ .

**Step 3.** If  $\|s\|_2 \leq \Delta$ ,

**Step 3a.** If  $\lambda = 0$  or  $\|s\|_2 = \Delta$ , stop.

**Step 3b.** Otherwise, compute an eigenvector  $u_1$  corresponding to  $\lambda_1$ , find the root  $\alpha$  of the equation  $\|s + \alpha u_1\|_2 = \Delta$  which makes the model  $q(s + \alpha u_1)$  smallest, replace  $s$  by  $s + \alpha u_1$ , and stop.

**Step 4.** If

$$|\|s\|_2 - \Delta| \leq \kappa_{\text{easy}} \Delta,$$

stop.

**Step 5.** Solve  $Lw = s$  and replace  $\lambda$  by  $\lambda + \left( \frac{\|s\|_2 - \Delta}{\Delta} \right) \left( \frac{\|s\|_2^2}{\|w\|_2^2} \right)$ .

**Step 6.** Factorize  $H(\lambda) = LL^T$ , solve  $LL^T s = -g$ , and go to Step 4.

The value of the perturbation  $\lambda_1^+$  is clearly crucial. From a practical point of view, one would like it not only to be sufficiently different from  $\lambda_1$  so that the numerical  $LL^T$  factors of  $H(-\lambda_1^+)$  exist, and so that the global Q-linear rate of convergence of the Newton iteration in the easy case is reasonable, but also so that the hard case is properly identified. We shall return to Algorithm 7.3.6 in Section 7.5.4.

## Notes and References for Section 7.3

The bulk of the material in this section is a distillation of the papers of Hebden (1973), Gay (1981), Moré and Sorensen (1983), Sorensen (1982a), and Moré (1983). The use of the secular equation in the optimization context is due to Hebden (1973), but was previously suggested by Reinsch (1971) in conjunction with methods for spline approximation. The character of the solution when  $g = 0$  was first observed by Goldfeldt, Quandt, and Trotter (1966), while

that for the general hard case was noted by Sorensen (1982a). The LINPACK method is due to Cline et al. (1979) and appeared in the software package of the same name. It has subsequently been refined by Cline, Conn, and Van Loan (1982). See also Higham (1987). Algorithm 7.3.4, and most of the associated technical results, is essentially that of Moré and Sorensen (1983), which builds on those of Moré (1978) and Gay (1981). This algorithm was implemented as GQTPAR, as part of the MINPACK software package, by Moré and Sorensen (1983). Software for unconstrained minimization based on such methods include UNCMIN by Schnabel, Koontz, and Weiss (1985), TRIDI by Dennis et al. (1991), and VA21 from the Harwell Subroutine Library.

Algorithm 7.3.6 is a generalization of that proposed in the tridiagonal case by Gould, Lucidi, Roma, and Toint (1999). Dennis et al. (1991) and Powell (1998c) proposed using a decomposition of the form (7.3.30) to obtain a band matrix to allow for efficient solution of the trust-region subproblem. Powell shows how to update such a decomposition efficiently following a low-rank change to  $H$ , while Dennis et al. give a special secant update that preserves the form of the decomposition (7.3.30) when  $\bar{H}$  is tridiagonal.

Gander (1981), Golub and von Matt (1991), and Chan, Olkin, and Cooley (1992) have all proposed appropriate methods for the special case involving least-squares models,  $q(s) = \|As - b\|_2^2$ . Such problems also arise when regularizing ill-posed problems.

## 7.4 The Scaled $\ell_2$ -Norm Problem

Rather than simply using the  $\ell_2$  norm, the geometry of the underlying problem might suggest that it is better to measure distances in the related *M norm*

$$\|x\|_M \equiv \sqrt{\langle x, Mx \rangle},$$

where  $M$  is a symmetric positive definite matrix. The underlying trust-region subproblem is then

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \\ & \text{subject to} && \|s\|_M \leq \Delta. \end{aligned} \tag{7.4.1}$$

Fortunately, it is easy in theory to handle the *M*-norm subproblem using the tools we have just developed for the  $\ell_2$  norm. For, as  $M$  is positive definite, it may be expressed<sup>96</sup> as

$$M = S^{-T} S^{-1}$$

for some invertible matrix  $S$ ; for instance,  $S$  may be the inverse of the Cholesky factor of  $M$ . Writing  $\bar{s} = S^{-1}s$ , it is then easy to transform (7.4.1) to the equivalent  $\ell_2$ -norm problem

$$\begin{aligned} & \underset{\bar{s} \in \mathbb{R}^n}{\text{minimize}} && \bar{q}(\bar{s}) \equiv \langle \bar{g}, \bar{s} \rangle + \frac{1}{2} \langle \bar{s}, \bar{H}\bar{s} \rangle \\ & \text{subject to} && \|\bar{s}\|_2 \leq \Delta, \end{aligned} \tag{7.4.2}$$

---

<sup>96</sup>We choose this decomposition of  $M$  for compatibility with Section 6.7.

where

$$\bar{g} = S^T g \quad \text{and} \quad \bar{H} = S^T H S. \quad (7.4.3)$$

The solution  $s^M$  to (7.4.1) may subsequently be recovered from the solution,  $\bar{s}^M$ , of (7.4.2) via the relationship

$$s^M = S\bar{s}^M. \quad (7.4.4)$$

All of the  $\ell_2$ -norm results in Sections 7.2 and 7.3.1 then have analogs for the  $M$ -norm subproblem. For instance, Corollary 7.2.2 becomes the following theorem.

**Theorem 7.4.1** Any global minimizer of  $q(s)$  subject to  $\|s\|_M \leq \Delta$  satisfies the equation

$$H(\lambda^M)s^M = -g, \quad (7.4.5)$$

where  $H(\lambda^M) \equiv H + \lambda^M M$  is positive semidefinite,  $\lambda^M \geq 0$ , and  $\lambda^M(\|s^M\|_M - \Delta) = 0$ . If  $H(\lambda^M)$  is positive definite,  $s^M$  is unique.

**Proof.** Applying Corollary 7.2.2 to (7.4.2), we see that the solution  $\bar{s}^M$  satisfies the equation

$$(\bar{H} + \lambda I)\bar{s}^M = -\bar{g}.$$

This equation yields (7.4.5) under the transformations (7.4.3) and (7.4.4). The inertia of  $H + \lambda M$  and  $\bar{H} + \lambda I$  are the same since

$$\bar{H} + \lambda I = S^T H S + \lambda I = S^T (H + \lambda S^{-T} S^{-1}) S = S^T (H + \lambda M) S$$

for the nonsingular factor  $S^{-1}$  of  $M$ . □

Notice how it is the matrix  $H + \lambda M$  rather than  $H + \lambda I$  that plays the central role for the  $M$ -norm subproblem. Simple modifications of Algorithm 7.3.4 or 7.3.6 may be used to find the model minimizer of (7.4.2); all that changes is that  $H(\lambda)$  is  $H + \lambda M$  rather than  $H + \lambda I$ ,  $\lambda_1$  and  $u_1$  are generalized eigensolutions to  $(H, M)$ , and the  $M$  norm appears rather than the  $\ell_2$  norm.

## 7.5 Approximating the Model Minimizer

While the method described in Section 7.3.1 is most definitely effective, it is not necessarily efficient. By this, we mean that the cost per iteration of Algorithm 7.3.4 is often dominated by the factorization of  $H(\lambda)$ . This is especially so if the problem is large, unless  $H$  has a favourable sparsity pattern. In this section, we consider alternative methods that aim to avoid the high overhead of computing a series of factorizations while approximating the model minimizer.

### 7.5.1 The Truncated Conjugate Gradient Method

In Section 5.1, we considered the conjugate gradient method for finding the minimizer of a strictly convex quadratic function. We mentioned there that it is possible to use the method for nonconvex problems, and that is what we shall describe here.

If we do not know whether our quadratic model is strictly convex, we must take precautions to deal with nonconvexity if it arises. Certainly, if the model is unbounded from below, our only safeguard is the trust region. On the other hand, if the model turns out to be convex, we should not interfere with the normal behaviour of the conjugate gradient method.

In Section 5.1.6, we stressed that in practice we usually prefer the preconditioned version of the conjugate gradient method. In any case, the normal algorithm is simply a special case of this more general method, and thus in this section it is convenient to consider the preconditioned case. As we observed in Section 6.7, preconditioning is simply a rescaling of the variables and thus a redefinition of the shape of the trust region. If  $M$  is our symmetric, positive definite preconditioner, the trust-region problem we consider is the  $M$ -norm problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \quad \text{subject to} \quad \|s\|_M \leq \Delta. \quad (7.5.1)$$

Suppose we were to apply Algorithm 5.1.4 (p. 88) to the minimization of  $q(s)$  regardless of whether or not  $H$  is positive definite or whether or not the generated iterates lie within the trust region. Then a number of possibilities might arise. Firstly, the curvature  $\langle p_k, Hp_k \rangle$  may be positive at each iteration while the iterates<sup>97</sup>  $s_k$  all remain within the trust region. This corresponds to the convex interior-solution case discussed in Section 7.2. Secondly, it may happen that  $\langle p_k, Hp_k \rangle \leq 0$  at iteration  $k$ . In this case, the model is not strictly convex, and the stepsize  $\alpha_k$  as computed from (5.1.55) (p. 88) does not give a reduction in  $q$ ; indeed,  $q(s)$  is unbounded from below along the line  $s_k + \alpha p_k$ . If our aim is to minimize the model within the trust region, it makes far more sense to reduce  $q$  along  $s_k + \alpha p_k$  as much as we can while staying within the trust region, and this means moving to the trust-region boundary along this line. Thus when  $\langle p_k, Hp_k \rangle \leq 0$ , we replace (5.1.55) with the positive root<sup>98</sup> of

$$\|s_k + \alpha p_k\|_M = \Delta. \quad (7.5.2)$$

The third possibility is that  $s_k$  lies outside the trust region at iteration  $k$ . At first sight, it is not obvious what to do now, as it is conceivable that subsequent iterates may re-enter the trust region, and thus interfering with the current wayward iterate might spoil the expected convergence behaviour of the preconditioned conjugate gradient method. However, this is not the case.

---

<sup>97</sup>The iterates here are denoted  $s_k$ , not  $x_k$ , to reflect that it is  $s$  rather than  $x$  we are seeking from Algorithm 5.1.4.

<sup>98</sup>The positive of the two roots gives the smaller value of  $q(s_k + \alpha p_k)$ .

**Theorem 7.5.1** Suppose that Algorithm 5.1.4 (p. 88)—the preconditioned conjugate gradient method—is applied to minimize  $q(s)$  starting from  $s_0 = 0$ , and that  $\langle p_i, H p_i \rangle > 0$  for  $0 \leq i \leq k$ . Then the iterates  $s_j$  satisfy the inequalities

$$\|s_j\|_M < \|s_{j+1}\|_M$$

for  $0 \leq j \leq k - 1$ .

**Proof.** We first show that

$$\langle p_j, M p_i \rangle = \frac{\langle g_j, v_j \rangle}{\langle g_i, v_i \rangle} \langle p_i, M p_i \rangle > 0 \quad (7.5.3)$$

for all  $0 \leq i \leq j \leq k$ . For any  $i$ , (7.5.3) is trivially true for  $j = i$ . Suppose it is also true for all  $i \leq l$ . Then, (5.1.57) and (5.1.58) (p. 88) give

$$p_{l+1} = -v_{l+1} + \frac{\langle g_{l+1}, v_{l+1} \rangle}{\langle g_l, v_l \rangle} p_l.$$

Forming the inner product with  $M p_i$  and using (5.1.59) (p. 88) and (7.5.3) when  $j = l$  reveals that

$$\begin{aligned} \langle p_{l+1}, M p_i \rangle &= -\langle v_{l+1}, M p_i \rangle + \frac{\langle g_{l+1}, v_{l+1} \rangle}{\langle g_l, v_l \rangle} \langle p_l, M p_i \rangle \\ &= \frac{\langle g_{l+1}, v_{l+1} \rangle}{\langle g_l, v_l \rangle} \frac{\langle g_l, v_l \rangle}{\langle g_i, v_i \rangle} \langle p_i, M p_i \rangle \\ &= \frac{\langle g_{l+1}, v_{l+1} \rangle}{\langle g_i, v_i \rangle} \langle p_i, M p_i \rangle. \end{aligned}$$

As  $M$  is positive definite, each  $\langle g_l, v_l \rangle > 0$  (using (5.1.56), p. 88), and thus (7.5.3) is true for  $i \leq l + 1$  and hence for all  $0 \leq i \leq j \leq k$ .

We now have from the algorithm that

$$s_j = s_0 + \sum_{i=0}^{j-1} \alpha_i p_i = \sum_{i=0}^{j-1} \alpha_i p_i,$$

as, by assumption,  $s_0 = 0$ . Hence

$$\langle s_j, M p_j \rangle = \left\langle \sum_{i=0}^{j-1} \alpha_i p_i, M p_j \right\rangle = \sum_{i=0}^{j-1} \alpha_i \langle p_i, M p_j \rangle > 0 \quad (7.5.4)$$

as each  $\alpha_i > 0$ , which follows from the definition (5.1.55) (p. 88) because of the assumption that  $\langle p_i, H p_i \rangle > 0$ , and from relationship (7.5.3). Hence

$$\begin{aligned} \|s_{j+1}\|_M^2 &= \langle s_{j+1}, M s_{j+1} \rangle \\ &= \langle s_j + \alpha_j p_j, M (s_j + \alpha_j p_j) \rangle \\ &= \langle s_j, M s_j \rangle + 2\alpha_j \langle s_j, M p_j \rangle + \alpha_j^2 \langle p_j, M p_j \rangle \\ &> \langle s_j, M s_j \rangle \\ &= \|s_j\|_M^2 \end{aligned}$$

follows directly from (7.5.4) and  $\alpha_j > 0$ , which is the required result.  $\square$

Thus we see that, so long as only positive curvature is encountered in the preconditioned conjugate gradient method, the  $M$  norm of the iterates strictly increases *provided* that the method is started with  $s_0 = 0$ . If  $H$  is positive definite, the iterates increase in the  $M$  norm, and thus

$$\|s_j\|_M \leq \|\arg \min_{s \in \mathbb{R}^n} q(s)\|_M,$$

with the inequality here being strict except at the terminating iteration. In particular, when  $H$  is positive definite and  $s_k$  lies outside the trust region, the solution to the trust-region problem must lie on the trust-region boundary. Thus, there is no reason to continue with the conjugate gradient iteration, as it stands, as subsequent iterates will move further outside the trust-region boundary. A sensible strategy, just as in the second case considered above, is to move to the trust-region boundary by finding the positive root of (7.5.2).

To summarize, if we apply the preconditioned conjugate gradient method to the minimization of  $q(s)$ , starting from  $s_0 = 0$ , regardless of whether or not  $H$  is positive definite or whether or not the generated iterates lie within the trust region, there can be three outcomes. Firstly, if  $H$  is positive definite, the method may converge to the convex interior solution. Secondly, if negative curvature is encountered, the method can be modified so that a useful point on the trust-region boundary is found. Finally, if the iterates leave the trust region, they will not return, so there is no virtue in following them, and again a useful point on the trust-region boundary is found. Once a point on the boundary has been generated, we know that the model minimizer is not interior. As there is no obvious further use for the conjugate gradient iteration, the simplest strategy is to stop as soon as the boundary is reached. We illustrate these possibilities in Figure 7.5.1, where the piecewise linear conjugate gradient path is shown in bold.

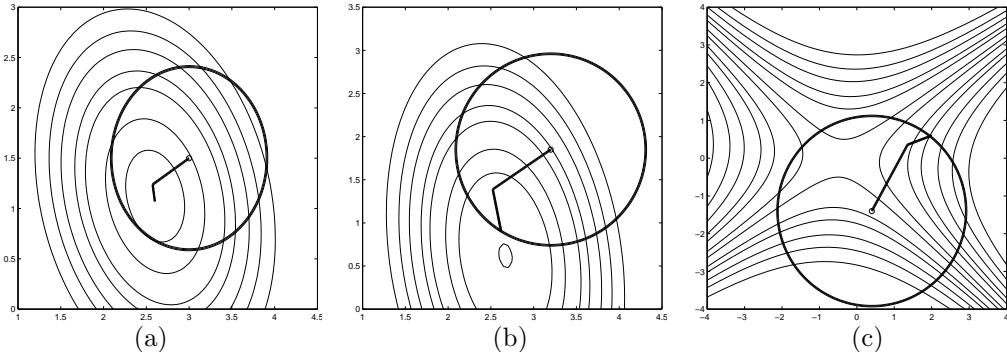


Figure 7.5.1: How the Steihaug–Toint method may terminate. (a) An interior solution is found. (b) The model is convex, but the solution lies outside the trust region, and the method terminates where the path crosses the boundary. (c) The model is nonconvex, the path has a segment of negative curvature, and the method terminates where the path crosses the boundary on this segment.

This is the basis of the Steihaug–Toint truncated conjugate gradient method.

**Algorithm 7.5.1: The Steihaug–Toint truncated conjugate gradient method**

Let  $s_0 = 0$ ,  $g_0 = g$ ,  $v_0 = M^{-1}g_0$ , and  $p_0 = -v_0$ . For  $k = 0, 1, \dots$  until convergence, perform the iteration:

```

Set  $\kappa_k = \langle p_k, H p_k \rangle$ .
If  $\kappa_k \leq 0$ ,
    compute  $\sigma_k$  as the positive root of  $\|s_k + \sigma p_k\|_M = \Delta$ , set
     $s_{k+1} = s_k + \sigma_k p_k$ , and
    stop.
End if
Set  $\alpha_k = \langle g_k, v_k \rangle / \kappa_k$ .
If  $\|s_k + \alpha_k p_k\|_M \geq \Delta$ ,
    compute  $\sigma_k$  as the positive root of  $\|s_k + \sigma p_k\|_M = \Delta$ .
    Set  $s_{k+1} = s_k + \sigma_k p_k$ , and
    stop.
End if
Set  $s_{k+1} = s_k + \alpha_k p_k$ ,
 $g_{k+1} = g_k + \alpha_k H p_k$ ,
 $v_{k+1} = M^{-1}g_{k+1}$ ,
 $\beta_k = \langle g_{k+1}, v_{k+1} \rangle / \langle g_k, v_k \rangle$ , and
 $p_{k+1} = -v_{k+1} + \beta_k p_k$ .
```

What can one say about the terminating iterate from Algorithm 7.5.1? Quite remarkably, so long as the condition number of  $M$  remains bounded over the sequence of subproblems approximately solved by Algorithm BTR (p. 116), then *any* iterate generated by Algorithm 7.5.1 is sufficient to ensure convergence to a first-order critical point. That this is so results from the fact that the *first* iterate,  $s_1$ , generated by Algorithm 7.5.1 is the Cauchy point for the model, and subsequent iterates give lower values of the model. Thus assumption AA.1 holds at each  $s_k$  for  $k > 0$ . See Section 6.7 for further details. Notice that it is the serendipitous choice of  $r_0 = g_0$  in the conjugate gradient method that makes this all possible.

At each stage of the Steihaug–Toint method we need to calculate  $\|s_k + \alpha_k p_k\|_M$ . This is not an issue if  $M$  is available. However, it may be the case that all that is actually available is a procedure which returns  $M^{-1}v$  for a given input  $v$ , and thus  $M$  is unavailable. Fortunately this is not a significant drawback as it is possible to calculate  $\|s_k + \alpha_k p_k\|_M$  from available information.

To see this, observe that

$$\|s_k + \alpha p_k\|_M^2 = \|s_k\|_M^2 + 2\alpha \langle s_k, Mp_k \rangle + \alpha^2 \|p_k\|_M^2$$

and that the positive root of  $\|s_k + \sigma p_k\|_M = \Delta$  is given by

$$\sigma_k = \frac{-\langle s_k, Mp_k \rangle + \sqrt{\langle s_k, Mp_k \rangle^2 + \|p_k\|_M^2(\Delta^2 - \|s_k\|_M^2)}}{\|p_k\|_M^2}. \quad (7.5.5)$$

Thus we can calculate this root and find  $\|s_{k+1}\|_M^2$  from  $\|s_k\|_M^2$  so long as we already know  $\langle s_k, Mp_k \rangle$  and  $\|p_k\|_M^2$ . But it is straightforward to show that these quantities may be calculated from the pair of recurrences (the second already appeared as (5.1.60), p. 88)

$$\langle s_k, Mp_k \rangle = \beta_{k-1} (\langle s_{k-1}, Mp_{k-1} \rangle + \alpha_{k-1} \|p_{k-1}\|_M^2) \quad \text{and} \quad (7.5.6)$$

$$\|p_k\|_M^2 = \langle g_k, v_k \rangle + \beta_{k-1}^2 \|p_{k-1}\|_M^2, \quad (7.5.7)$$

where, of course,  $\langle g_k, v_k \rangle$  has already been calculated as part of the truncated conjugate gradient method. The key to (7.5.6) is the identity  $s_k = \sum_{i=0}^{k-1} \alpha_i p_i$ , which follows from the recursion for  $s_k$  in Algorithm 7.5.1, and (5.1.59) (p. 88).

There remains one further issue, namely, when to terminate Algorithm 7.5.1. Clearly, this is not a problem when the algorithm detects negative curvature or when an iterate exceeds the trust region, as the algorithm as it stands will terminate when either of these events occurs. In theory, if  $H$  is positive definite and  $s^M$  is interior, Corollary 5.1.9 (p. 89) indicates that the preconditioned conjugate gradient method will find  $s^M$  within a fixed number of iterations. However, it might well be expensive to wait for this promised convergence, and a less-expensive alternative is desirable.

The simplest truncation rule is to stop the algorithm after a fixed number of iterations. While it is virtually impossible to know in practice just how many iterations are necessary to provide a good estimate of  $s^M$ , almost all termination tests include such a rule as a last resort. A more popular rule is to stop when the norm of the gradient,  $\|g_k\|$ , or the preconditioned gradient,  $\|v_k\|$ , have been reduced to a small fraction of their initial values. While one should be cautious here, as this does not necessarily imply that  $s_k$  is a better estimate of  $s^M$  than was  $s_0$ , this rule performs adequately in practice. In order to improve the convergence rate of the underlying trust-region algorithm, it is normally necessary to solve the trust-region subproblem to increasingly higher accuracy as the trust-region algorithm approaches a first-order critical point. Thus the reduction in the gradient or preconditioned gradient is frequently required to approach zero as the underlying trust-region algorithm approaches criticality. A typical rule, which includes all of these ingredients, is to stop as soon as an iteration  $k$  is reached for which

$$\|g_k\| \leq \|g_0\| \min [\kappa_{\text{fgr}}, \|g_0\|^\theta] \quad \text{or} \quad k > k_{\max},$$

where<sup>99</sup>  $\kappa_{\text{fgr}} < 1$ ,  $\theta \geq 0$ , and  $k_{\max} \geq 0$ . So long as  $\theta > 0$  and a suitable model Hessian is

---

<sup>99</sup>“fgr” stands for “fraction of the gradient required”.

used, superlinear convergence of the underlying trust-region method is possible. Values like  $\kappa_{\text{fgr}} = 0.1$ ,  $\theta = 0.5$ , and  $k_{\max} = n$  are typical.

Another possible but, in our opinion, unappealing technique is to increase the accuracy requirement on each successive subproblem. A common theme here is to include a stopping rule based on the index of the outer iteration  $\ell$  to stop, for instance, when  $\|g_k\|$  or  $\|v_k\|$  is smaller than  $1/\ell$ . We dislike such a rule both because of its seeming arbitrariness and because it is insensitive to the actual requirements of the model.

## Notes and References for Subsection 7.5.1

The first suggestion that the conjugate gradient method might be used to find an approximation to  $s^M$  came from Toint (1981b). The main difference between Toint's method and Algorithm 7.5.1, which is due to Steihaug (1983), is that the former retreats to the Cauchy point if negative curvature is detected. Theorem 7.5.1 is due to Steihaug.

The idea of truncating the solution of the subproblem while still attaining a fast convergence rate was first proposed by Dembo, Eisenstat, and Steihaug (1982) and Dembo and Steihaug (1983) in the absence of a trust region. Both Toint and Steihaug's proposals include such termination rules, and Steihaug shows that the underlying trust-region method will converge superlinearly, under modest assumptions, for a variety of model Hessians. See also Eisenstat and Walker (1994). Nash (1985) suggests ways of computing effective preconditioners, while Nash and Sofer (1990) warn that basing termination rules solely on the size of the (preconditioned) gradient may be misleading and give examples showing that poor points may be accepted with such rules.

The truncated conjugate gradient method is not the only proposal in which an approximate trust-region step is obtained by restricting the problem to a suitable subspace. A number of authors, including Bulteau and Vial (1983, 1985, 1987), Byrd, Schnabel, and Shultz (1987, 1988), Martínez (1997), Branch (1995), Branch, Coleman, and Li (1999), L. P. Sun (1996), Bouaricha and Schnabel (1997), Coleman and Li (1998a), and Heinkenschloss (1998), consider restricting the subproblem to a two-dimensional subspace. Typically this subspace is chosen as that spanned by the steepest-descent direction and either the step to the unconstrained model minimizer, if it is convex, or a direction of sufficient negative curvature otherwise. Note that, in the latter case, this amounts to minimizing the model in the subspace spanned by the Cauchy step and the step to the eigenpoint (see Chapter 6).

It is also possible to apply the truncated conjugate gradient method when the solution to the  $M$ -norm subproblem (7.5.1) is additionally constrained to satisfy a set of affine constraints  $As = 0$ . The key is to observe, as we did in Section 5.4.1, that the preconditioned conjugate gradient method may be performed implicitly within the manifold  $As = 0$ ; this gave us the projected preconditioned conjugate gradient method, Algorithm 5.4.2 (p. 109). So long as the preconditioner and the trust-region norm are consistent—by this, we mean that the matrix  $M$  used in the preconditioning step (5.4.5) (p. 110) is the same  $M$  that defines the trust-region norm—the projected preconditioned conjugate gradient method will generate iterates that all satisfy  $As = 0$  and that increase in the  $M$  norm. Thus, Algorithm 7.5.1 may be applied directly to

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \quad \text{subject to } As = 0 \text{ and } \|s\|_M \leq \Delta \quad (7.5.8)$$

so long as we replace the preconditioning steps  $v_{k+1} = M^{-1}g_{k+1}$  by solutions of the linear system (5.4.5).

### 7.5.2 How Good Is the Steihaug–Toint Point?

We now examine the quality of the point produced by the Steihaug–Toint truncated conjugate gradient method as an approximation of the model minimizer. A satisfactory answer can be derived in the case where the model is convex. This subsection is devoted to showing that the model reduction obtained at the point produced by the truncated conjugate gradient method,  $s^{\text{ST}}$ , is at least half of that obtained at the model minimizer,  $s^M$ , so long as the only truncation rules allowed are those stated explicitly in Algorithm 7.5.1 (that is, that the trust-region boundary is encountered) or that  $\langle g_k, v_k \rangle = 0$  for some index  $k$ . We first derive some further properties of Algorithm 7.5.1. For simplicity, we consider here its unpreconditioned version; that is, we shall assume (for the remainder of this subsection) that  $M = I$ .

**Lemma 7.5.2** Let  $\{p_k\}$  be the sequence of search directions produced by Algorithm 7.5.1 with  $M = I$  and let  $\{s_k\}$  be the corresponding sequence of trial steps. Then, for any  $k \geq 0$  such that  $g_k \neq 0$ , we have that

$$p_k = -\|g_k\|^2 \sum_{i=0}^k \frac{g_i}{\|g_i\|^2} \quad (7.5.9)$$

and

$$s_{k+1} = -\sum_{i=0}^k \frac{g_i}{\|g_i\|^2} \sum_{j=i}^k \alpha_j \|g_j\|^2. \quad (7.5.10)$$

**Proof.** By definition,  $p_0 = -g_0 = -g$  and (7.5.9) thus holds for  $k = 0$ . Assume therefore that it holds for  $i = 0, \dots, k-1$ . The definition of the algorithm gives that

$$\begin{aligned} p_k &= -g_k + \beta_{k-1} p_{k-1} \\ &= -g_k + \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \left( -\|g_{k-1}\|^2 \sum_{i=0}^{k-1} \frac{g_i}{\|g_i\|^2} \right) \\ &= -\|g_k\|^2 \left( \frac{g_k}{\|g_k\|^2} + \sum_{i=0}^{k-1} \frac{g_i}{\|g_i\|^2} \right) \\ &= -\|g_k\|^2 \sum_{i=0}^k \frac{g_i}{\|g_i\|^2}. \end{aligned}$$

Thus (7.5.9) holds by induction for all  $k \geq 0$  provided  $g_k \neq 0$ . Moreover, the algorithm ensures that

$$s_{k+1} = \sum_{j=0}^k \alpha_j p_j = -\sum_{j=0}^k \alpha_j \|g_j\|^2 \sum_{i=0}^j \frac{g_i}{\|g_i\|^2} = -\sum_{i=0}^k \frac{g_i}{\|g_i\|^2} \sum_{j=i}^k \alpha_j \|g_j\|^2,$$

which is (7.5.10). □

Since minimizing  $\langle g, s \rangle + \frac{1}{2}\langle s, Hs \rangle$  is the same as minimizing

$$\langle Q^T g, Qs \rangle + \frac{1}{2}\langle Qs, (QHQ^T)Qs \rangle$$

for any orthonormal matrix  $Q$ , we see that the conjugate gradient algorithm is invariant to orthonormal transformation. Since  $g$  must<sup>100</sup> lie in the range of  $H$ , we thus assume for the rest of this subsection, and without loss of generality, that we choose  $Q$  as an orthonormal matrix whose first column is  $g/\|g\|$  and such that

$$QHQ^T = T = \begin{pmatrix} T_m & 0 \\ 0 & 0 \end{pmatrix},$$

where  $T_m$  is an  $m \times m$  positive definite tridiagonal matrix ( $m \leq n$ ). This is exactly what we did in Section 5.2.6 (with  $m = n$ ) when making the connection between the conjugate gradient procedure and the Lanczos algorithm. We thus restrict our attention to the case where

$$H = T \text{ is positive semidefinite and } g = \|g\|e_1 = \gamma_0 e_1 \quad (7.5.11)$$

(see (5.2.53) [p. 104] and (5.2.35) [p. 99]). As in Chapter 5, we also denote by  $T_k$  the  $(k+1) \times (k+1)$  leading principal submatrix of  $T$  ( $k \leq m$ ), that is,

$$T_k = \begin{pmatrix} \delta_0 & \gamma_1 & & & \\ \gamma_1 & \delta_1 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{pmatrix}$$

(see (5.2.15) [p. 95]). Using this notation, we deduce from (5.2.36) (p. 99) that

$$s_{k+1} = -\|g\| \begin{pmatrix} T_k^{-1} e_1 \\ 0 \end{pmatrix}, \quad (7.5.12)$$

where  $e_1$  is interpreted as the first vector in the canonical basis of  $\mathbb{R}^{k+1}$ .

We also derive another expression for the gradient of the  $(k+1)$ st model.

**Lemma 7.5.3** If  $\gamma_i \neq 0$  for  $i = 1, \dots, k$ , then

$$g_{k+1} = (-1)^{k-1} \|g\| \frac{\prod_{i=1}^{k+1} \gamma_i}{\det(T_k)} e_{k+2} \quad (7.5.13)$$

and

$$\|g_{k+1}\|^2 = \|g\|^2 \frac{\prod_{i=1}^{k+1} \gamma_i^2}{[\det(T_k)]^2}. \quad (7.5.14)$$

Moreover,  $g_{k+1} = 0$  for  $k < m$  if and only if  $\gamma_{k+1} = 0$ .

<sup>100</sup>Indeed,  $g = H(0 - x_*)$ , where  $x_*$  is the (unique) minimizer of the model in the range of  $H$ .

**Proof.** We have that

$$g_{k+1} = g + Ts_{k+1} = -\|g\| \begin{bmatrix} -e_1 + T \begin{pmatrix} T_k^{-1} e_1 \\ 0 \end{pmatrix} \end{bmatrix} = -\|g\| \gamma_{k+1} e_1^T T_k^{-1} e_{k+1} e_{k+2}, \quad (7.5.15)$$

where we have exploited the tridiagonal form of  $T$ . If  $k = 0$ , this gives that

$$g_1 = -\|g\| \gamma_1 e_1^T T_0^{-1} e_1 e_2 = -\|g\| \frac{\gamma_1}{\det(T_0)} e_2,$$

which is (7.5.13). For  $k > 0$ , the tridiagonal form of  $T_k$  implies that<sup>101</sup>

$$e_1 T_k^{-1} e_{k+1} = (-1)^k \frac{\prod_{i=1}^k \gamma_i}{\det(T_k)},$$

which, combined with (7.5.15), yields (7.5.13). Equation (7.5.14) immediately follows from (7.5.13), as does the last conclusion of the lemma.  $\square$

We will return to results of this type in Section 7.5.4. We also note that, if  $g_{k+1} = 0$  for some  $k < n - 1$ , we may restrict the problem to the subspace spanned by the first  $k + 1$  canonical basis vectors. Hence there is no loss of generality in assuming that  $\gamma_i \neq 0$  for  $i = 0, \dots, m$ . We now parametrize the path produced by the truncated conjugate gradient method by defining

$$s(t) = s_k + (t - k)(s_{k+1} - s_k) \text{ for } t \in [k, k + 1].$$

We may then describe this trajectory componentwise.

**Lemma 7.5.4** We have that, for all  $t \in [0, k + 1]$ ,

$$s(t) = - \sum_{i=0}^k \varrho_i(t) \operatorname{sgn}(\langle e_{i+1}, g_i \rangle) e_{i+1} \quad (7.5.16)$$

where, for  $i = 0, \dots, k$ , we define  $\varrho_i(t) = 0$  for  $t \in [0, i]$  and

$$\varrho_i(t) = \frac{1}{\|g_i\|} \left( \sum_{j=i}^{\lfloor t \rfloor - 1} \alpha_j \|g_j\|^2 + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor} \|g_{\lfloor t \rfloor}\|^2 \right) \quad (7.5.17)$$

for  $t \in [i, k + 1]$ .

---

<sup>101</sup>This relationship may be obtained from the well-known expression of the inverse of a matrix as the transpose of the cofactor matrix divided by the determinant (see Noble and Daniel, 1977, Theorem 6.9), the observation that  $e_1^T T_k^{-1} e_k$  is nothing other than element  $(1, k)$  of the inverse of  $T_k$ , and the fact that element  $(1, k)$  of the cofactor is the determinant of a triangular matrix having the  $\gamma_i$ 's on its diagonal.

**Proof.** Using the mechanism of the algorithm, (7.5.9), (7.5.10), and the convention that an empty sum is zero, we obtain that

$$\begin{aligned}
s(t) &= s_{\lfloor t \rfloor} + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor} p_{\lfloor t \rfloor} \\
&= - \sum_{i=0}^{\lfloor t-1 \rfloor} \frac{g_i}{\|g_i\|^2} \sum_{j=i}^{\lfloor t-1 \rfloor} \alpha_j \|g_j\|^2 + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor} \left( -\|g_{\lfloor t \rfloor}\|^2 \sum_{i=0}^{\lfloor t \rfloor} \frac{g_i}{\|g_i\|^2} \right) \\
&= - \sum_{i=0}^{\lfloor t-1 \rfloor} \frac{g_i}{\|g_i\|^2} \sum_{j=i}^{\lfloor t-1 \rfloor} \alpha_j \|g_j\|^2 - \frac{g_{\lfloor t \rfloor}}{\|g_{\lfloor t \rfloor}\|^2} \sum_{j=\lfloor t \rfloor}^{\lfloor t-1 \rfloor} \alpha_j \|g_j\|^2 \\
&\quad + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor} \left( -\|g_{\lfloor t \rfloor}\|^2 \sum_{i=0}^{\lfloor t \rfloor} \frac{g_i}{\|g_i\|^2} \right) \\
&= - \sum_{i=0}^{\lfloor t \rfloor} \frac{g_i}{\|g_i\|^2} \left( \sum_{j=i}^{\lfloor t-1 \rfloor} \alpha_j \|g_j\|^2 + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor} \|g_{\lfloor t \rfloor}\|^2 \right) \\
&= - \sum_{i=0}^{\lfloor t \rfloor} \frac{g_i}{\|g_i\|} \varrho_i(t) \\
&= - \sum_{i=0}^k \frac{g_i}{\|g_i\|} \varrho_i(t).
\end{aligned}$$

Now, the identity (7.5.13) gives that  $g_i$  is a multiple of  $e_{i+1}$ , and hence that

$$\operatorname{sgn}(\langle e_{i+1}, g_i \rangle) e_{i+1} = \frac{g_i}{\|g_i\|},$$

and we therefore obtain the desired conclusion by substituting this last relation in the penultimate equation.  $\square$

We are now in position to express the slope along the successive trial steps and the norm of these steps in more detail.

**Lemma 7.5.5** The coefficients  $\{\varrho_i(t)\}_{i=0}^k$  are strictly increasing functions of  $t$  for  $t \in [i, k+1]$ . Furthermore, we have that, for  $t \in [0, k+1]$ ,

$$-\langle g, s(t) \rangle = \|g\| \varrho_0(t), \tag{7.5.18}$$

$$\|s(t)\|^2 = \sum_{i=0}^k [\varrho_i(t)]^2, \tag{7.5.19}$$

and the functions  $-\langle g, s(t) \rangle$  and  $\|s(t)\|$  are strictly increasing in this interval.

**Proof.** From the second part of (7.5.11) and (7.5.16), we deduce that

$$-\langle g, s(t) \rangle = -\|g\| \langle e_1, s(t) \rangle = \|g\| \varrho_0(t),$$

which proves (7.5.18). We also deduce (7.5.19) from (7.5.16). The desired monotonicity conclusions then follow from the observation that  $\varrho_i(t)$  must be strictly

increasing for  $t \in [i, k+1]$  because of (7.5.17).  $\square$

Note that this lemma provides an alternative proof of Theorem 7.5.1 for  $M = I$ . The identity (7.5.18) provides a useful equivalence between  $\varrho_0(t)$  and  $-\langle g, s(t) \rangle$ , which we systematically exploit in what follows.

The next step in our reasoning is to consider the minimization of the “shifted model”

$$q(s, \lambda) \stackrel{\text{def}}{=} \langle g, s \rangle + \frac{1}{2} \langle s, H(\lambda)s \rangle,$$

where  $H(\lambda)$  is defined in Theorem 7.2.1 for some  $\lambda \geq 0$  (instead of  $q(s) = \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle$ ), and to parametrize the results obtained above as a function of  $\lambda$ . From (7.5.13), we immediately deduce that, for  $k > 0$ ,

$$g_{k+1}(\lambda) = \frac{\det(T_k)}{\det(T_k + \lambda I)} g_{k+1}, \quad (7.5.20)$$

while  $g_0(\lambda) = g_0 = g$ . Moreover, the *shifted* truncated conjugate gradient path is now given, for  $t \in [k, k+1]$ , by

$$s(t, \lambda) = s_k(\lambda) + (t - k)(s_{k+1}(\lambda) - s_k(\lambda)) = -\sum_{i=0}^k \varrho_i(t, \lambda) \operatorname{sgn}(\langle e_{i+1}, g_i(\lambda) \rangle) e_{i+1},$$

where for  $i = 0, \dots, k$ , we define  $\varrho_i(t) = 0$  for  $t \in [0, i]$  and

$$\varrho_i(t, \lambda) = \frac{1}{\|g_i(\lambda)\|} \left( \sum_{j=i}^{\lfloor t \rfloor - 1} \alpha_j(\lambda) \|g_j(\lambda)\|^2 + (t - \lfloor t \rfloor) \alpha_{\lfloor t \rfloor}(\lambda) \|g_{\lfloor t \rfloor}(\lambda)\|^2 \right) \quad (7.5.21)$$

for  $t \in [i, k+1]$ . Moreover, we may deduce from the definition of  $s(t, \lambda)$  and (7.5.18) applied to the shifted model that

$$-\langle g, s_k(\lambda) \rangle = -\langle g, s(k, \lambda) \rangle < -\langle g, s(k+1, \lambda) \rangle = -\langle g, s_{k+1}(\lambda) \rangle \quad (7.5.22)$$

for  $k \geq 0$ . We may now start comparing the shifted quantities with the unshifted ones.

**Lemma 7.5.6** For any  $k \geq 0$  and for any  $\lambda > 0$  such that  $g_k(\lambda) \neq 0$ , we have that

$$-\langle g, s_{k+1}(\lambda) \rangle < -\langle g, s_{k+1} \rangle, \quad (7.5.23)$$

$$\|s_{k+1}(\lambda)\| < \|s_{k+1}\|, \quad (7.5.24)$$

and, if  $k > 0$ ,

$$\|g_k(\lambda)\| < \|g_k\|. \quad (7.5.25)$$

**Proof.** Using (7.5.11), (7.5.12), and the positive definite nature of  $T_k$ , we obtain that

$$-\langle g, s_{k+1}(\lambda) \rangle = \|g\|^2 \langle e_1, (T_k + \lambda I)^{-1} e_1 \rangle < \|g\|^2 \langle e_1, T_k^{-1} e_1 \rangle = -\langle g, s_{k+1} \rangle$$

and

$$\|s_{k+1}(\lambda)\| = \|g\|^2 \langle e_1, (T_k + \lambda I)^{-2} e_1 \rangle < \|g\|^2 \langle e_1, T_k^{-2} e_1 \rangle = \|s_{k+1}\|,$$

which proves (7.5.23) and (7.5.24). Now also deduce from (7.5.20) and the positive definite nature of  $T_{k-1}$  that

$$\|g_k(\lambda)\| = \frac{\det(T_{k-1})}{\det(T_{k-1} + \lambda I)} \|g_k\| < \|g_k\| \quad (7.5.26)$$

and our proof is complete.  $\square$

Note that (7.5.26) implies that  $g_k(\lambda)$  and  $g_k$  are always nonzero together (if  $\lambda$  is finite). Our next lemma provides the crucial technical step.

**Lemma 7.5.7** For any  $k \geq 0$  such that  $g_k(\lambda) \neq 0$  and for any  $\lambda > 0$ , there exists a  $t \in [0, k+1]$  such that

$$-\langle g, s_{k+1}(\lambda) \rangle = -\langle g, s(t) \rangle \quad (7.5.27)$$

and

$$\varrho_i(t) \leq \varrho_i(k+1, \lambda) \quad (7.5.28)$$

for  $i = 0, \dots, k$ . Moreover, this last inequality is strict when  $k > 0$  and  $i > 0$ .

**Proof.** The proof is by induction on the iteration number, which we denote  $\ell$ . Consider first the case where  $\ell = 0$ . The definitions (7.5.17) and (7.5.21) give that, for  $t \in [0, 1]$ ,

$$\varrho_0(t) = t\alpha_0\|g\| \text{ and } \varrho_0(1, \lambda) = \alpha_0(\lambda)\|g\|. \quad (7.5.29)$$

But

$$\alpha_0(\lambda) = \frac{\|g\|^2}{\langle g, T_0(\lambda)g \rangle} < \frac{\|g\|^2}{\langle g, T_0g \rangle} = \alpha_0$$

since  $T_0$  is positive definite. Combining this relation with (7.5.29), we obtain that

$$\varrho_0(t) = \varrho_0(1, \lambda)$$

for

$$t = \frac{\langle g, T_0g \rangle}{\langle g, T_0(\lambda)g \rangle} \in (0, 1),$$

which gives (7.5.28) for  $\ell = 0$ . We then derive (7.5.27) for  $\ell = 0$  from (7.5.18) applied to both the shifted and unshifted versions of the trust-region subproblem.

Now assume that there exists a  $t \in [0, \ell+1]$  such that

$$-\langle g, s_{\ell+1}(\lambda) \rangle = -\langle g, s(t) \rangle \quad (7.5.30)$$

and

$$\varrho_i(t) \leq \varrho_i(\ell+1, \lambda) \quad (7.5.31)$$

for  $i = 0, \dots, \ell$ . If  $g_{\ell+1} \neq 0$ , we deduce from (7.5.30), (7.5.22), and (7.5.23) that

$$-\langle g, s(t) \rangle = -\langle g, s_{\ell+1}(\lambda) \rangle < -\langle g, s_{\ell+2}(\lambda) \rangle < -\langle g, s_{\ell+2} \rangle = -\langle g, s(\ell+2) \rangle.$$

But, since  $-\langle g, s(t) \rangle$  is both continuous and, because of Lemma 7.5.5, increasing, we obtain that there exists a  $\bar{t} \in (t, \ell+2) \subset [0, \ell+2)$  such that

$$-\langle g, s_{\ell+2}(\lambda) \rangle = -\langle g, s(\bar{t}) \rangle, \quad (7.5.32)$$

which is (7.5.27) for  $\ell+2$ . Note that  $\bar{t} < \ell+2$  because of (7.5.23). Observe now that, because of (7.5.21),

$$\varrho_i(\ell+2, \lambda) = \varrho_i(\ell+1, \lambda) + \alpha_{\ell+1}(\lambda) \frac{\|g_{\ell+1}(\lambda)\|^2}{\|g_i(\lambda)\|}$$

for  $i = 0, \dots, \ell+1$ , which yields that

$$\frac{\varrho_i(\ell+2, \lambda) - \varrho_i(\ell+1, \lambda)}{\varrho_0(\ell+2, \lambda) - \varrho_0(\ell+1, \lambda)} = \frac{\|g\|}{\|g_i(\lambda)\|} \quad (7.5.33)$$

for  $i = 0, \dots, \ell+1$ . Observe also that

$$\begin{aligned} \varrho_i(\bar{t}) &= \varrho_i(t) \\ &+ \frac{1}{\|g_i\|} \left( (\lfloor t \rfloor + 1 - t) \alpha_{\lfloor t \rfloor} \|g_{\lfloor t \rfloor}\|^2 + \sum_{j=\lfloor t \rfloor+1}^{\lfloor \bar{t} \rfloor-1} \alpha_j \|g_j\|^2 + (\bar{t} - \lfloor \bar{t} \rfloor) \alpha_{\lfloor \bar{t} \rfloor} \|g_{\lfloor \bar{t} \rfloor}\|^2 \right) \end{aligned}$$

for  $i = 0, \dots, \lfloor t \rfloor$ , and

$$\varrho_i(\bar{t}) = \varrho_i(t) + \frac{1}{\|g_i\|} \left( \sum_{j=i}^{\lfloor \bar{t} \rfloor-1} \alpha_j \|g_j\|^2 + (\bar{t} - \lfloor \bar{t} \rfloor) \alpha_{\lfloor \bar{t} \rfloor} \|g_{\lfloor \bar{t} \rfloor}\|^2 \right)$$

for  $i = \lfloor t \rfloor + 1, \dots, \lfloor \bar{t} \rfloor$ . Combining the above relations, we deduce that

$$\frac{\varrho_i(\bar{t}) - \varrho_i(t)}{\varrho_0(\bar{t}) - \varrho_0(t)} \leq \frac{\|g\|}{\|g_i\|}$$

for  $i = 0, \dots, \lfloor \bar{t} \rfloor$ . Therefore we obtain that, for  $i = \lfloor t \rfloor + 1, \dots, \lfloor \bar{t} \rfloor$ ,

$$\varrho_i(\bar{t}) = \varrho_i(t) + [\varrho_i(\bar{t}) - \varrho_i(t)] \leq \varrho_i(t) + \frac{\|g\|}{\|g_i\|} [\varrho_0(\bar{t}) - \varrho_0(t)]. \quad (7.5.34)$$

Now

$$\|g\| \varrho_0(\bar{t}) = -\langle g, s(\bar{t}) \rangle = -\langle g, s_{\ell+2}(\lambda) \rangle = -\langle g, s(\ell+2, \lambda) \rangle = \|g\| \varrho_0(\ell+2, \lambda) \quad (7.5.35)$$

because of (7.5.18) and (7.5.32). Similarly,

$$\|g\| \varrho_0(t) = -\langle g, s(t) \rangle = -\langle g, s_{\ell+1}(\lambda) \rangle = -\langle g, s(\ell+1, \lambda) \rangle = \|g\| \varrho_0(\ell+1, \lambda) \quad (7.5.36)$$

because of (7.5.18) and (7.5.30). Thus (7.5.34) gives that, for  $i = 1, \dots, \min[\lfloor \bar{t} \rfloor, \ell]$ ,

$$\begin{aligned}\varrho_i(\bar{t}) &\leq \varrho_i(t) + \frac{\|g\|}{\|g_i\|} [\varrho_0(\ell+2, \lambda) - \varrho_0(\ell+1, \lambda)] \\ &= \varrho_i(t) + \frac{\|g_i(\lambda)\|}{\|g_i\|} [\varrho_i(\ell+2, \lambda) - \varrho_i(\ell+1, \lambda)] \\ &< \varrho_i(t) + [\varrho_i(\ell+2, \lambda) - \varrho_i(\ell+1, \lambda)] \\ &< \varrho_i(\ell+2, \lambda),\end{aligned}$$

where we have used (7.5.33), (7.5.25), and (7.5.31) successively. This inequality and the fact that  $\varrho_i(\bar{t}) = 0$  for  $i > \lfloor \bar{t} \rfloor$  imply that

$$\varrho_i(\bar{t}) < \varrho_i(\ell+2, \lambda) \quad (7.5.37)$$

for  $i = 1, \dots, \ell$ . Now, if  $\bar{t} \leq \ell+1$ , then  $\varrho_{\ell+1}(\bar{t}) = 0$ , and thus

$$\varrho_{\ell+1}(\bar{t}) < \varrho_{\ell+1}(\ell+2, \lambda). \quad (7.5.38)$$

Otherwise, that is, if  $\bar{t} \in (\ell+1, \ell+2)$ , we have that

$$\begin{aligned}\varrho_{\ell+1}(\bar{t}) &= \varrho_{\ell+1}(\bar{t}) - \varrho_{\ell+1}(\ell+1) \\ &= \frac{\|g\|}{\|g_{\ell+1}\|} [\varrho_0(\bar{t}) - \varrho_0(\ell+1)] \\ &< \frac{\|g\|}{\|g_{\ell+1}\|} [\varrho_0(\bar{t}) - \varrho_0(t)] \\ &= \frac{\|g\|}{\|g_{\ell+1}\|} [\varrho_0(\ell+2, \lambda) - \varrho_0(\ell+1, \lambda)] \\ &= \frac{\|g\|}{\|g_{\ell+1}\|} \frac{\|g_{\ell+1}(\lambda)\|}{\|g_\ell\|} [\varrho_{\ell+1}(\ell+2, \lambda) - \varrho_{\ell+1}(\ell+1, \lambda)] \\ &< \varrho_{\ell+1}(\ell+2, \lambda) - \varrho_{\ell+1}(\ell+1, \lambda) \\ &= \varrho_{\ell+1}(\ell+2, \lambda).\end{aligned} \quad (7.5.39)$$

We have used the identity  $\varrho_{\ell+1}(\ell+1) = 0$ , the definition (7.5.17), the fact that  $t \leq \ell+1$ , the monotone nature of  $\varrho_0(t)$ , (7.5.35), (7.5.36), (7.5.33), (7.5.25), and the identity  $\varrho_{\ell+1}(\ell+1, \lambda) = 0$ . Our induction argument is then completed by combining (7.5.37), (7.5.38), and (7.5.39), and the lemma is proved.  $\square$

Our final lemma identifies the norm of a trial step on the truncated conjugate gradient path with that of an iterate on the shifted model.

**Lemma 7.5.8** For any  $g \geq 0$  such that  $g_k(\lambda) \neq 0$  and for any  $\lambda > 0$ , there exists a unique  $\hat{t} \in [0, k+1)$  such that

$$\|s(\hat{t})\| = \|s_{k+1}(\lambda)\| \quad (7.5.40)$$

and

$$-\langle g, s_{k+1}(\lambda) \rangle \leq -\langle g, s(\hat{t}) \rangle. \quad (7.5.41)$$

Moreover, this last inequality is strict when  $k > 0$ .

**Proof.** From the preceding lemma, we know that there exists a  $t \in [0, k+1)$  such that (7.5.27) and (7.5.28) hold, with this last inequality being strict for  $k > 0$  and  $i > 0$ . Observe now that (7.5.27) and (7.5.18) imply that

$$\varrho_0(t) = \varrho_0(k+1, \lambda),$$

which, combined with (7.5.28) and (7.5.19), gives that

$$\|s_{k+1}(\lambda)\| = \|s(k+1, \lambda)\| \geq \|s(t)\|, \quad (7.5.42)$$

the inequality being strict for  $k > 0$ . On the other hand,

$$\|s(k+1)\| = \|s_{k+1}\| > \|s_{k+1}(\lambda)\|$$

because of (7.5.24). Combining these bounds and using the monotonically increasing nature of  $\|s(t)\|$  implied by Lemma 7.5.5, we obtain that there must exist a  $\hat{t} \in [t, k+1)$  such that

$$\|s(\hat{t})\| = \|s_{k+1}(\lambda)\|.$$

Because  $\hat{t} \geq t$ , we also have that

$$-\langle g, s(\hat{t}) \rangle \geq -\langle g, s(t) \rangle = -\langle g, s_{k+1}(\lambda) \rangle, \quad (7.5.43)$$

where we used the strictly monotone nature of  $\langle g, s(t) \rangle$ , which follows from Lemma 7.5.5 and from (7.5.27). If  $k > 0$ , inequality (7.5.42) is strict and therefore so is the inequality in (7.5.43), giving the desired conclusion.  $\square$

After these substantial preliminaries, we are now at last in position to state the main result of this subsection.

**Theorem 7.5.9** Assume that  $q(s)$  is convex and that the Steihaug–Toint truncated conjugate gradient method, Algorithm 7.5.1, stops only if the boundary is encountered or if  $\nabla_s q(s) = 0$ . If this algorithm is applied to compute  $s^{\text{ST}}$ , an approximation to  $s^M$ , the exact solution of the trust-region subproblem of minimizing  $q(s)$  for  $\|s\| \leq \Delta$ , then

$$q(s^{\text{ST}}) \leq \frac{1}{2}q(s^M). \quad (7.5.44)$$

**Proof.** Let  $\lambda = \lambda^M$  be the solution of the secular equation associated with the trust-region subproblem. If  $\lambda = 0$ , then the minimizer of the model in the range of  $H$  is interior to the trust region and there exists a  $k \leq m$

$$s^M = s_{k+1}(\lambda) = s^{ST},$$

in which case the desired bound trivially follows. Assume now that  $\lambda > 0$ . Then the solution is on the boundary of the trust region, and there exists a  $k \leq m$  such that  $g_k(\lambda) \neq 0$ ,  $g_{k+1}(\lambda) = 0$ , and hence

$$s^M = s_{k+1}(\lambda). \quad (7.5.45)$$

Lemma 7.5.8 then ensures that there exists a  $\hat{t} \in [0, k+1]$  such that (7.5.40) and (7.5.41) hold. Furthermore

$$\|s(\hat{t})\| = \|s_{k+1}(\lambda)\| = \Delta,$$

and we therefore conclude from the mechanisms of Algorithm 7.5.1, as well as the strictly increasing nature of  $\|s(t)\|$  implied by Lemma 7.5.5, that

$$s(\hat{t}) = s^{ST}. \quad (7.5.46)$$

Observe now that  $q(\alpha s^{ST})$  is a convex quadratic in  $\alpha$  whose minimizer is

$$\alpha_* = -\frac{\langle g, s^{ST} \rangle}{\langle s^{ST}, Hs^{ST} \rangle} \geq 1.$$

Therefore, we have that

$$q(\alpha_* s^{ST}) = \alpha_* \langle g, s^{ST} \rangle + \frac{1}{2} \alpha_*^2 \langle s^{ST}, Hs^{ST} \rangle = \frac{1}{2} \alpha_* \langle g, s^{ST} \rangle$$

and

$$q(\alpha s^{ST}) \leq \frac{1}{2} \alpha \langle g, s^{ST} \rangle \quad (7.5.47)$$

for all  $\alpha \in [0, \alpha_*]$ . As a consequence, we obtain that

$$q(s^{ST}) \leq \frac{1}{2} \langle g, s^{ST} \rangle = \frac{1}{2} \langle g, s(\hat{t}) \rangle \leq \frac{1}{2} \langle g, s_{k+1}(\lambda) \rangle = \frac{1}{2} \langle g, s^M \rangle \leq \frac{1}{2} q(s^M),$$

where we have successively used (7.5.47) for  $\alpha = 1$ , (7.5.46), (7.5.41), (7.5.45), the definition of  $q(s)$ , and the inequality  $\langle s^M, Hs^M \rangle \geq 0$ .  $\square$

Not only is this result interesting because it ensures that the Steihaug–Toint method is all we need to obtain an acceptable approximation, in the sense of (7.3.19), to the model minimizer in the convex case, but it is also somewhat surprising because the bound does not depend on the eigenvalues of  $H$ . Of course, the main limitation of Theorem 7.5.9 is that checking that a model is convex essentially requires that we know or calculate (suitable bounds on) the eigenvalues of  $H$ , and the latter may require significant computation when the problem involves a large number of unknowns.

An extension of this result to the nonconvex case is unfortunately impossible. Indeed, if  $g = 0$  and  $H$  is indefinite, the method will terminate at  $s = 0$  with no decrease in the model, while the model minimizer will occur on the trust-region boundary along a direction of most negative curvature. We will return to methods that avoid this defect in Section 7.5.4.

We conclude this analysis of the quality of the approximation produced by the Steihaug–Toint method by noting that Theorem 7.5.9 remains valid if the preconditioned version of the algorithm is considered. Indeed, the value of the model is invariant for any nonsingular and symmetric change of variables, that is,

$$q(s) = \langle M^{-1}g, Ms \rangle + \frac{1}{2}\langle Ms, (M^{-1}HM^{-1})Ms \rangle$$

for any  $s$ , and thus (7.5.44) is unaffected by preconditioning.

## Notes and References for Subsection 7.5.2

Theorem 7.5.9 and its preparatory lemmas are due to Yuan (2000). This paper provides a formal proof, in the strictly convex case, of the conjecture made by Yuan (1997), where the result of Theorem 7.5.9 was shown in the special case where  $n = 2$  and the model is strictly convex. The (simple) extension to the general convex case appears to be new. We are also aware of work by P. Tseng (private communication, September 1998) where the factor of  $\frac{1}{2}$  in the statement of this theorem is replaced by  $\frac{1}{3}$ .

### 7.5.3 Dogleg and Double-Dogleg Paths

An earlier approach to the trust-region subproblem is to replace the search for an approximate solution over the current Krylov space with a search over a much simpler domain, namely, a path  $s(\alpha)$  for  $0 \leq \alpha \leq \alpha_{\max}$  and some suitable  $\alpha_{\max}$ . The path is required to start from the origin when  $\alpha = 0$ , to connect a number of strategic points or *knots*, and to be such that  $q(s(\alpha))$  decreases monotonically as  $\alpha$  increases. The aim, then, is to solve the problem

$$\underset{0 \leq \alpha \leq \alpha_{\max}}{\text{minimize}} \quad q(s(\alpha)) \quad \text{subject to} \quad \|s(\alpha)\|_M \leq \Delta. \quad (7.5.48)$$

The simplest such methods use linear segments to join the knots, and of these the most widely used are the dogleg method of Powell and the double-dogleg method of Dennis and Mei. For both of these, suppose that  $s^C$  is a gradient-related Cauchy point (in the sense of Section 8.1.5),  $s^I$  and  $s^J$  are “at least as good” directions in the sense that

$$q(s^J) < q(s^I) \leq q(s^C) \quad \text{and} \quad \|s^C\|_M \leq \|s^I\|_M < \|s^J\|_M,$$

and  $q(s)$  decreases along the line starting from  $s^I$  and ending at  $s^J$ . For instance, if we apply the preconditioned conjugate gradient method (Algorithm 5.1.4 [p. 88]) to minimize  $q$  starting from  $s_0 = 0$ , then any pair of generated iterates  $s_i$  and  $s_j$  suffices for  $s^I$  and  $s^J$  when  $1 \leq i < j$  and  $q(s)$  is convex. This follows because  $s_1$  is

gradient-related, each  $s_k$  minimizes  $q$  over the subspace spanned by  $\{s_1, \dots, s_k\}$ , and Theorem 7.5.1 shows that  $\|s_i\|_M < \|s_j\|_M$  if  $i < j$ . In particular, if  $i = 1$  and  $j = n$ ,  $s^I$  and  $s^J$  are, respectively, the Cauchy and Newton directions

$$s^I = -\frac{\langle g, M^{-1}g \rangle}{\langle g, Hg \rangle} M^{-1}g \quad \text{and} \quad s^J = -H^{-1}g.$$

This is the choice suggested by Powell, and Dennis and Mei, in the convex case.

The dogleg method involves two knots, the points  $s^I$  and  $s^J$ . The resulting *dogleg path* is then

$$s(\alpha) = \begin{cases} \alpha s^I & \text{for } 0 \leq \alpha \leq 1, \\ s^I + (\alpha - 1)(s^J - s^I) & \text{for } 1 \leq \alpha \leq 2. \end{cases}$$

The double-dogleg method adds an extra knot between  $s^I$  and  $s^J$ , at the point  $\gamma s^J$  for some carefully chosen  $0 < \gamma \leq 1$ . The *double-dogleg path* is then given by

$$s(\alpha) = \begin{cases} \alpha s^I & \text{for } 0 \leq \alpha \leq 1, \\ s^I + \frac{\alpha - 1}{\gamma}(\gamma s^J - s^I) & \text{for } 1 \leq \alpha \leq 1 + \gamma, \\ (\alpha - 1)s^J & \text{for } 1 + \gamma \leq \alpha \leq 2. \end{cases}$$

The value  $\gamma$  is chosen to ensure that  $q(s(\alpha))$  and  $\|s(\alpha)\|_M$ , respectively, decrease and increase monotonically for  $1 \leq \alpha \leq 1 + \gamma$ . It is easy to see that this is possible because of our requirement that  $q(s)$  decrease along the straight line starting from  $s^I$  and ending at  $s^J$  and because  $\|s^I\|_M < \|s^J\|_M$ . We illustrate the two paths in Figure 7.5.2.

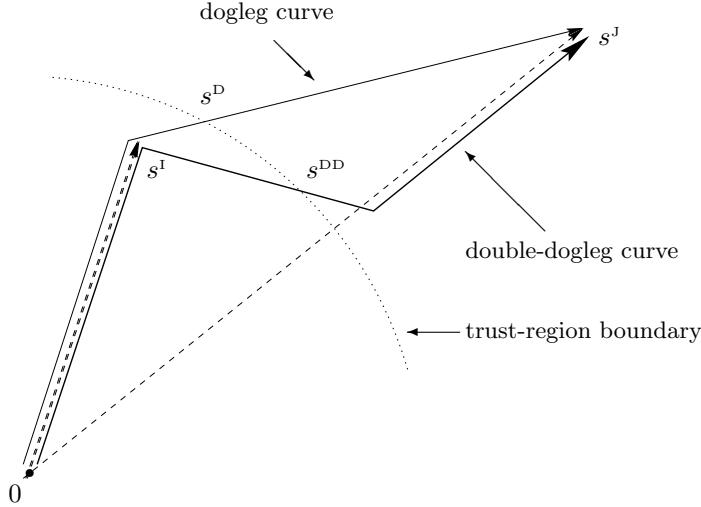


Figure 7.5.2: The dogleg and double-dogleg paths. The points  $s^D$  and  $s^{DD}$  indicate where the two paths leave the trust region.

As was the case in Section 7.5.1, the convergence of Algorithm BTR (p. 116) to a first-order critical point using the step generated by the dogleg or double-dogleg methods is easy to establish so long as the matrices  $H$  and  $M$  are positive definite with

uniformly bounded condition numbers. Again, this is a consequence of the behaviour of  $q$  along the segment  $0 \leq \alpha \leq 1$  of  $s(\alpha)$ . Let  $\alpha^M$  be the minimizing value of (7.5.48). If  $0 \leq \alpha^M < 1$ ,

$$\Delta = \|s(\alpha^M)\|_M = \alpha^M \|s^I\|_M < \|s^I\|_M,$$

and thus the Cauchy point  $s^C = s(\alpha^M) = \alpha^M s^I$  is the minimizing point. If, on the other hand,  $\alpha^M \geq 1$ , the entire segment  $0 \leq \alpha \leq 1$  of  $s(\alpha)$  lies within the trust region. Hence

$$q(s(\alpha^M)) < q(s^I) \leq q(s^C),$$

and thus  $s(\alpha^M)$  gives a smaller model value than at a gradient-related Cauchy point. Therefore AA.1 holds at  $s(\alpha^M)$  and convergence to a first-order critical point is ensured.

It is perhaps worth mentioning that one can also view the Steihaug–Toint method, Algorithm 7.5.1, as a search along the piecewise linear path  $s(\alpha)$ , whose  $i$ th piece is the segment

$$s_{i-1} + (\alpha - i + 1)(s_i - s_{i-1}), \quad \text{where } i-1 \leq \alpha \leq i,$$

and where the path ends as it leaves the trust region or when an interior solution is found. Thus such methods can trace their ancestry back to the dogleg idea, and have all but superseded the latter.

The cost of computing the (double-)dogleg curve is usually dominated by that of obtaining the Newton direction from a matrix factorization. Notice that this is usually less than the cost of Algorithm 7.3.4, where more than one factorization may be required, but typically more than that for the truncated conjugate gradient scheme given in Algorithm 7.5.1. The restriction that the model be convex is a clear disadvantage of the (double-)dogleg approach as we have described it. Powell’s original method was designed for the nonconvex models, and in this case  $s^J$  is chosen to be on the segment

$$s^C + \beta(-H^{-1}g - s^C)$$

for some appropriate (possibly negative)  $\beta$ . This and other nonconvex dogleg methods have much in common with the methods we shall consider in the next section.

### Notes and References for Subsection 7.5.3

The dogleg method is due to Powell (1970a, 1970d), while the double-dogleg variant was proposed by Dennis and Mei (1979), the latter resulting in the Fortran subroutines MINOP and MINEW. Kaufman (1999) extends this approach to allow for limited-memory quasi-Newton Hessian approximations.<sup>102</sup> Reid (1973) compares the performance of the dogleg method with the exact solution of the subproblem. More general paths, along which the model can be minimized within the trust region, have been analysed by Amaya (1985), Bulteau and Vial (1987), Nabona (1987), Xiao and Zhou (1992), McCartin (1998), Zhang, Xu, and Du (1998), and Zhang and Xu (1999a, 1999c).

---

<sup>102</sup>See Section 8.4.1.2.

Several authors have also suggested that the search for an approximate solution of the trust-region subproblem might be restricted to a low-dimensional subspace, as such an approximation may often be computed cheaply. As for the dogleg strategy, this subspace is often chosen to contain the steepest-descent direction and either an approximate Newton direction, when the model is convex, or some direction of sufficient negative curvature otherwise. This approach was advocated by Bulteau and Vial (1985) and Shultz, Schnabel, and Byrd (1985) and properly evaluated for the first time by Byrd, Schnabel, and Shultz (1988). This strategy has been further analysed by Branch (1995), Branch, Coleman, and Li (1999), Heinkenschloss (1998), L. P. Sun (1996), and Williamson (1990). We have already seen this idea in Section 7.5.1, and will pursue it further in Section 7.5.4, in which we will explore the possibility of nested subspaces.

### 7.5.4 The Truncated Lanczos Approach

The Steihaug–Toint method considered in Section 7.5.1 is basically unconcerned with the trust region until it blunders into its boundary and stops. This is rather unfortunate, particularly as considerable experience has shown that this frequently happens during the first few, and often the first, iteration(s) when negative curvature is present. The resulting step is then barely, if at all, better than the Cauchy direction, and this leads to a slow but globally convergent algorithm in theory and a barely convergent method in practice. In this section, we consider an alternative that aims to avoid this drawback by trying harder to solve the subproblem when the boundary is encountered, but maintains the efficiencies of the conjugate gradient method so long as the iterates lie interior to the trust region. The mechanism we use is the Lanczos method.

To set the scene, we recall from (6.3.3) (p. 124) and (6.7.6)/(6.7.7) (p. 164) that the Cauchy point (in the norm defined by the preconditioner  $M$ ) may be defined as the solution to the problem

$$\underset{s \in \text{span}\{M^{-1}g\}}{\text{minimize}} \quad q(s) \equiv \langle g, s \rangle + \frac{1}{2}\langle s, Hs \rangle \quad \text{subject to} \quad \|s\|_M \leq \Delta.$$

That is, the Cauchy point is the minimizer of  $q$  within the trust region where  $s$  is restricted to the one-dimensional subspace  $\text{span}\{M^{-1}g\}$ . The dogleg methods aim to solve the same problem over a one-dimensional path. In both of these cases the solution is easy to find, as the search space is small. The difficulty with the general problem (7.5.1) is that the search space  $\mathbb{R}^n$  is large. This leads immediately to the possibility of solving a compromise problem

$$\underset{s \in \mathcal{S}}{\text{minimize}} \quad q(s) \quad \text{subject to} \quad \|s\|_M \leq \Delta, \quad (7.5.49)$$

where  $\mathcal{S}$  is a specially chosen subspace of  $\mathbb{R}^n$ .

Now consider the Steihaug–Toint method at an iteration  $k$  before the trust-region boundary is encountered. In this case, the point  $s_{k+1}$  is the solution to (7.5.49) with the set

$$\mathcal{S} = \mathcal{K}(M^{-1}H, M^{-1}g, k). \quad (7.5.50)$$

That is, the Steihaug–Toint method is gradually widening the search space using the efficient preconditioned conjugate gradient method. However, as soon as the Steihaug–Toint method moves across the trust-region boundary, the terminating point  $s_{k+1}$  no longer necessarily solves the problem in (7.5.49) for  $S$  given by (7.5.50); indeed it is very unlikely to do so when  $k > 0$ . Can we do better? Yes, by recalling that the preconditioned conjugate gradient and Lanczos methods generate different bases for the same Krylov space and the special structure that ensues.

Rather than use the preconditioned conjugate gradient basis  $\{p_0, p_1, \dots, p_k\}$  for  $\mathcal{S}$ , we shall use the equivalent Lanczos  $M$ -orthonormal basis<sup>103</sup>  $\{q_0, q_1, \dots, q_k\}$ . Letting  $Q_k$  be the matrix  $(q_0 \cdots q_k)$ , the key relationships are

$$Q_k^T M Q_k = I_{k+1} \quad \text{and} \quad Q_k^T M q_{k+1} = 0, \quad (7.5.51)$$

$$H Q_k - M Q_k T_k = \gamma_{k+1} w_{k+1} e_{k+1}^T, \quad (7.5.52)$$

$$Q_k^T H Q_k = T_k, \quad \text{and} \quad (7.5.53)$$

$$Q_k^T g = \gamma_0 e_1, \quad (7.5.54)$$

which were (5.2.51)–(5.2.55) (p. 105) in Section 5.2.6, together with the elementary identity

$$g = M y_0 = \gamma_0 M q_0, \quad (7.5.55)$$

which follows from (5.2.46) and (5.2.48) (p. 104).

We shall consider vectors of the form

$$s \in \mathcal{S} = \{s \in \mathbb{R}^n \mid s = Q_k h\}$$

and seek

$$s_k = Q_k h_k, \quad (7.5.56)$$

where  $s_k$  solves the problem

$$\underset{s \in \mathcal{S}}{\text{minimize}} \quad q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, H s \rangle \quad \text{subject to} \quad \|s\|_M \leq \Delta. \quad (7.5.57)$$

It then follows directly from (7.5.51), (7.5.53), and (7.5.54) that  $h_k$  solves the problem

$$\underset{h \in \mathbb{R}^{k+1}}{\text{minimize}} \quad \langle h, \gamma_0 e_1 \rangle + \frac{1}{2} \langle h, T_k h \rangle \quad \text{subject to} \quad \|h\|_2 \leq \Delta. \quad (7.5.58)$$

There are a number of crucial observations to be made here. Firstly, it is important to note that the resulting trust-region problem involves the  $\ell_2$  rather than the  $M$  norm. Secondly, as  $T_k$  is tridiagonal, it is feasible to use Algorithm 7.3.4 to compute the model minimizer *even* when  $n$  is large. However, computing eigensolutions of tridiagonal matrices is also practical, and so the alternative, Algorithm 7.3.6, may be more appropriate. Thirdly, having found  $h_k$ , the matrix  $Q_k$  is needed to recover  $s_k$ , and thus the Lanczos vectors will either need to be saved on backing store or regenerated

---

<sup>103</sup>Two vectors  $p$  and  $q$  are  $M$ -orthogonal if  $\langle p, Mq \rangle = 0$ . They are  $M$ -orthonormal if additionally  $\langle p, Mp \rangle = \langle q, Mq \rangle = 1$ .

when required. As we shall see, we only need  $Q_k$  once we are satisfied that continuing the Lanczos process will give little extra benefit. Fourthly, one would hope that as a sequence of such problems may be solved, and as  $T_k$  only changes by the addition of an extra diagonal and superdiagonal entry, solution data from one subproblem may be useful for starting the next.

The basic trust-region solution classification theorem, Theorem 7.4.1, shows that

$$(T_k + \lambda_k I_{k+1})h_k = -\gamma_0 e_1, \quad (7.5.59)$$

where  $T_k + \lambda_k I_{k+1}$  is positive semidefinite,  $\lambda_k \geq 0$ , and  $\lambda_k(\|h_k\|_2 - \Delta) = 0$ . What does this tell us about  $s_k$ ? Firstly, using (7.5.53), (7.5.54), (7.5.56), and (7.5.59) we have

$$Q_k^T(H + \lambda_k M)s_k = Q_k^T(H + \lambda_k M)Q_k h_k = (T_k + \lambda_k I_{k+1})h_k = -\gamma_0 e_1 = -Q_k^T g,$$

and additionally that

$$\lambda_k(\|s_k\|_M - \Delta) = 0 \quad \text{and} \quad \lambda_k \geq 0. \quad (7.5.60)$$

Comparing these with the trust-region classification theorem, we see that  $s_k$  is the Galerkin approximation<sup>104</sup> to  $s^M$  from the space spanned by  $Q_k$ .

We may then ask how good the approximation is. In particular, what is the error  $(H + \lambda_k M)s_k + g$ ? The simplest way of measuring this error would be to calculate  $h_k$  and  $\lambda_k$  by solving (7.5.58), then to recover  $s_k$  as  $Q_k h_k$ , and finally to substitute  $s_k$  and  $\lambda_k$  into  $(H + \lambda_k M)s_k + g$ . However, this is inconvenient, as it requires that we have easy access to  $Q_k$ . Fortunately there is a far better way.

**Theorem 7.5.10**

$$(H + \lambda_k M)s_k + g = \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} \quad (7.5.61)$$

and

$$\|(H + \lambda_k M)s_k + g\|_{M^{-1}} = \gamma_{k+1}|\langle e_{k+1}, h_k \rangle|. \quad (7.5.62)$$

**Proof.** We have that

$$\begin{aligned} Hs_k &= HQ_k h_k && \text{from (7.5.56)} \\ &= MQ_k T_k h_k + \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} && \text{from (7.5.52)} \\ &= -MQ_k(\lambda_k h_k + \gamma_0 e_1) + \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} && \text{from (7.5.59)} \\ &= -\lambda_k MQ_k h_k - \gamma_0 MQ_k e_1 + \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} \\ &= -\lambda_k M s_k - \gamma_0 M q_0 + \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} \\ &= -\lambda_k M s_k - g + \gamma_{k+1}\langle e_{k+1}, h_k \rangle w_{k+1} && \text{from (7.5.55).} \end{aligned}$$

<sup>104</sup>Broadly speaking, the Galerkin approximation of the solution of a problem from a subspace is the solution of a variant of this problem restricted to this subspace. Such a variant may be defined, for instance, by a suitable projection or by discretization of an infinite-dimensional problem.

This then directly gives (7.5.61). Now  $w_i = Mq_i$  by definition, and the  $q_i$  are  $M$ -orthonormal. Hence,

$$\langle w_i, M^{-1}w_j \rangle = \langle Mq_i, q_j \rangle = \delta_{i,j}$$

for  $i, j = 1, \dots, k+1$ . The  $w_i$  are thus  $M^{-1}$ -orthonormal, and (7.5.62) follows.  $\square$

Thus we can indirectly measure the error (in the  $M^{-1}$  norm) knowing simply  $\gamma_{k+1}$  and the last component of  $h_k$ , and we do not need  $s_k$  or  $Q_k$  at all. Observant readers will notice the strong similarity between this error estimate and the estimate (5.2.54) (p. 105) for the gradient of the model in the Lanczos method, but this is not at all surprising, as the two methods are aiming for the same point if the trust-region radius is large enough. An interpretation of (7.5.62) is also identical to that of (5.2.54). The error will be small when either of  $\gamma_{k+1}$  or the last component of  $h_k$  is small.

We now consider the problem (7.5.58) in more detail. We say that a symmetric tridiagonal matrix is *reducible* if one or more of its off-diagonal entries is zero; otherwise it is *irreducible*. We then have the following preliminary result.

**Lemma 7.5.11** Suppose that the tridiagonal matrix  $T$  is irreducible and that  $v$  is an eigenvector of  $T$ . Then the first component of  $v$  is nonzero.

**Proof.** By definition,

$$Tv = \theta v \tag{7.5.63}$$

for some eigenvalue  $\theta$ . Suppose that the first component of  $v$  is zero. Considering the first component of (7.5.63), we have that the second component of  $v$  is zero as  $T$  is tridiagonal and irreducible. Repeating this argument for the  $i$ th component of (7.5.63), we deduce that the  $(i+1)$ st component of  $v$  is zero for all  $i$ , and hence that  $v = 0$ . But this contradicts the assumption that  $v$  is an eigenvector, and so the first component of  $v$  cannot be zero.  $\square$

This immediately yields the following useful result.

**Theorem 7.5.12** Suppose that  $T_k$  is irreducible. Then the hard case cannot occur for the subproblem (7.5.58).

**Proof.** Suppose the hard case occurs. Then, by definition,  $\gamma_0 e_1$  is orthogonal to  $v_k$ , the eigenvector corresponding to the leftmost eigenvalue,<sup>105</sup>  $-\theta_k$ , of  $T_k$ . Thus, the first component of  $v_k$  is zero, which, following Lemma 7.5.11, contradicts the assumption that  $v_k$  is an eigenvector. Thus the hard case cannot occur.  $\square$

---

<sup>105</sup>These eigenvalues are, of course, the Ritz values.

This result is important as it suggests that the full power of the Algorithm 7.3.4 is not needed to solve (7.5.58). We shall return to this at the end of this section. We also have an immediate corollary.

**Corollary 7.5.13** Suppose that  $T_{n-1}$  is irreducible. Then the hard case cannot occur for the original problem (7.5.1).

**Proof.** When  $k = n - 1$ , the columns of  $Q_{n-1}$  form a basis for  $\mathbb{R}^n$ . Thus the problems (7.5.1) and (7.5.57) are identical, and (7.5.57) and (7.5.58) are related through a nonsingular transformation. The result then follows directly from Theorem 7.5.12 in the case  $k = n - 1$ .  $\square$

Thus, if the hard case occurs for (7.5.1), the Lanczos tridiagonal must be reducible at some stage.

**Theorem 7.5.14** Suppose that  $T_k$  is irreducible, that  $h_k$  and  $\lambda_k$  satisfy (7.5.59), and that  $T_k + \lambda_k I_{k+1}$  is positive semidefinite. Then  $T_k + \lambda_k I_{k+1}$  is positive definite.

**Proof.** Suppose that  $T_k + \lambda_k I_{k+1}$  is singular. Then there is a nonzero eigenvector  $v_k$  for which  $(T_k + \lambda_k I_{k+1})v_k = 0$ . Hence, combining this with (7.5.59) reveals that

$$0 = \langle h_k, (T_k + \lambda_k I_{k+1})v_k \rangle = \langle v_k, (T_k + \lambda_k I_{k+1})h_k \rangle = -\gamma_0 \langle v_k, e_1 \rangle,$$

and hence that the first component of  $v_k$  is zero. But this contradicts Lemma 7.5.11. Hence  $T_k + \lambda_k I_{k+1}$  is both positive semidefinite and nonsingular, and thus positive definite.  $\square$

This result implies that (7.5.59) has a unique solution. We now consider this solution.

**Theorem 7.5.15** Suppose that  $\langle e_{k+1}, h_k \rangle = 0$ . Then  $T_k$  is reducible.

**Proof.** Suppose that  $T_k$  is irreducible. As the  $(k + 1)$ st component of  $h_k$  is zero, then from the nondegeneracy of  $T_k$  and the  $(k + 1)$ st equation of (7.5.59), we deduce that the  $k$ th component of  $h_k$  is zero. Repeating this argument for the  $(i + 1)$ st equation of (7.5.59), we deduce that the  $i$ th component of  $h_k$  is zero for  $1 \leq i \leq k$ , and hence that  $h_k = 0$ . But this contradicts the first equation of (7.5.59), and thus  $T_k$  must be reducible.  $\square$

Thus we see that of the two possibilities suggested by Theorem 7.5.10 for obtaining an  $s_k$  for which  $(H + \lambda_k M)s_k + g = 0$ , it will be the possibility  $\gamma_{k+1} = 0$  that occurs before  $\langle e_{k+1}, h_k \rangle = 0$ .

**Theorem 7.5.16** Suppose that the hard case does not occur for (7.5.1), and that  $\gamma_{k+1} = 0$ . Then  $s_k$  solves (7.5.1).

**Proof.** If  $\gamma_{k+1} = 0$ , (7.5.52) gives that  $HQ_k = MQ_kT_k$ . Now let  $T_k = U_kD_kU_k^T$  be the spectral decomposition of  $T_k$ . Then

$$M^{-1}HQ_k = Q_kT_k = Q_kU_kD_kU_k^T$$

and thus

$$(M^{-1}H)Q_kU_k = Q_kU_kD_k.$$

As a consequence, we see that the columns of  $Q_kU_k$  are eigenvectors of  $M^{-1}H$  associated with the eigenvalues contained in  $D_k$ . Since  $U_k$  is nonsingular as  $T_k$  is symmetric, we therefore deduce that the columns of  $Q_k$  span an invariant subspace of  $M^{-1}H$ . Hence, the Krylov space  $\mathcal{K}(M^{-1}H, M^{-1}g, k)$  is an invariant subspace of this matrix and, by construction, the first basis element of this space is  $M^{-1}g$ . As the hard case does not occur for (7.5.1), this space must also contain at least one eigenvector corresponding to the leftmost eigenvalue,  $-\theta$ , of  $M^{-1}H$ . Thus one of the eigenvalues of  $T_k$  must be  $-\theta$ , and  $\lambda_k \geq \theta$  as  $T_k + \lambda_k I_{k+1}$  is positive semidefinite. But this implies that  $H + \lambda_k M$  is positive semidefinite, which combines with (7.5.56), (7.5.60), and Theorem 7.5.10 with  $\gamma_{k+1} = 0$  to show that  $s_k$  satisfies the optimality conditions shown in Theorem 7.4.1.  $\square$

Thus we see that in the easy case, the required solution will be obtained from the first irreducible block of the Lanczos tridiagonal. It remains for us to consider the hard case. In view of Corollary 7.5.13, this case can only occur when  $T_k$  is reducible. Suppose therefore that  $T_k$  reduces into  $\ell$  blocks of the form

$$T_k = \begin{pmatrix} T_{k_1} & & & \\ & T_{k_2} & & \\ & & \ddots & \\ & & & T_{k_\ell} \end{pmatrix}, \quad (7.5.64)$$

where each of the  $T_{k_i}$  defines an invariant subspace for  $M^{-1}H$  and where the last block  $T_{k_\ell}$  is the first to yield the leftmost eigenvalue,  $-\theta$ , of  $M^{-1}H$ . Then there are two cases to consider.

**Theorem 7.5.17** Suppose that the hard case occurs for (7.5.1), that  $T_k$  is as described by (7.5.64), and that the last block  $T_{k_\ell}$  is the first to yield the leftmost eigenvalue,  $-\theta$ , of  $M^{-1}H$ . Then,

- (1) if  $\theta \leq \lambda_{k_1}$ , a solution to (7.5.1) is given by  $s_k = Q_{k_1}h_{k_1}$ , where  $h_{k_1}$  solves the positive definite system

$$(T_{k_1} + \lambda_{k_1} I_{k_1+1})h_{k_1} = -\gamma_0 e_1.$$

- (2) if  $\theta > \lambda_{k_1}$ , a solution to (7.5.1) is given by  $s_k = Q_k h_k$ , where

$$h_k = \begin{pmatrix} h \\ 0 \\ \vdots \\ 0 \\ \alpha u \end{pmatrix},$$

$h$  is the solution of the nonsingular tridiagonal system

$$(T_{k_1} + \theta I_{k_1+1})h = -\gamma_0 e_1,$$

$u$  is an eigenvector of  $T_{k_\ell}$  corresponding to  $-\theta$ , and  $\alpha$  is chosen so that

$$\|h\|_2^2 + \alpha^2 \|u\|_2^2 = \Delta^2.$$

**Proof.** In case (1),  $H + \lambda_{k_1} M$  is positive semidefinite as  $\lambda_{k_1} \geq \theta$ , and the remaining optimality conditions are satisfied as  $\gamma_{k_1+1} = 0$  and  $h_{k_1}$  solves (7.5.57). That  $T_{k_1} + \lambda_{k_1} I_{k_1+1}$  is positive definite follows from Theorem 7.5.14. In case (2),  $H + \theta M$  is positive semidefinite. Furthermore, as  $\theta > \lambda_{k_1}$ , it is easy to show that  $\|h\|_2 < \|h_{k_1}\|_2 \leq \Delta$ , and hence that there is a root  $\alpha$  for which  $\|s_k\|_M = \|h_k\|_2 = \Delta$ . Finally, as each  $Q_{k_i}$  defines an invariant subspace,  $HQ_{k_i} = MQ_{k_i}T_{k_i}$ . Writing  $s = Q_{k_1}h$  and  $v = Q_{k_\ell}u$ , we therefore have

$$Hs = HQ_{k_1}h = MQ_{k_1}T_{k_1}h = MQ_{k_1}(-\theta h - \gamma_0 e_1) = -\theta Ms - g$$

and

$$Hv = HQ_{k_\ell}u = MQ_{k_\ell}T_{k_\ell}u = -\theta MQ_{k_\ell}u = -\theta Mv.$$

Thus  $(H + \theta M)s_k = -g$ , and  $s_k$  satisfies all of the optimality conditions for (7.5.57).  $\square$

Notice that to obtain  $s_k$  as described in this theorem, we only require the Lanczos vectors corresponding to blocks 1 and, perhaps,  $\ell$  of  $T_k$ .

The reader should appreciate that to solve the problem as outlined in Theorem 7.5.17 may be unrealistic, as it relies on being sure that we have located the leftmost eigenvalue of  $M^{-1}H$ . With Lanczos-type methods, one cannot normally guarantee that

all eigenvalues, including the leftmost, will be found unless one ensures that all invariant subspaces have been investigated, and this may prove to be very expensive for large problems. In particular, the Lanczos algorithm, Algorithm 5.2.4 (p. 104), terminates each time an invariant subspace has been determined, and must be restarted using a vector  $q$  that is  $M$ -orthonormal to the previous Lanczos directions. Such a vector may be obtained from the Gram–Schmidt process by reorthonormalizing a suitable vector with respect to the previous Lanczos directions (a vector with some component  $M$ -orthogonal to the existing invariant subspaces or perhaps a random vector), which means that these directions will have to be regenerated or reread from external storage. Thus, while this form of the solution is of theoretical interest, it is unlikely to be of practical use if a cheap approximation to the solution is all that is required.

With this in mind, we now outline a realistic algorithm, Algorithm 7.5.2, the generalized Lanczos trust-region method. We stress that, as the goal is merely to improve upon the value delivered by the Steihaug–Toint method, we do not use the full power of Theorem 7.5.17 and are content just to investigate the first invariant subspace produced by the Lanczos algorithm. In almost all cases, this subspace contains the global solution to the problem, whereas the complications and costs required to implement a method based on Theorem 7.5.17 are likely prohibitive.

**Algorithm 7.5.2: The generalized Lanczos trust-region method**

Let  $s_0 = 0$ ,  $g_0 = g$ ,  $v_0 = M^{-1}g_0$ ,  $\gamma_0 = \sqrt{\langle v_0, g_0 \rangle}$ , and  $p_0 = -v_0$ . Set the flag **INTERIOR** as true. For  $k = 0, 1, \dots$  until convergence, perform the iteration:

```

Set  $\alpha_k = \langle g_k, v_k \rangle / \langle p_k, H p_k \rangle$ .
Obtain  $T_k$  from  $T_{k-1}$  using (5.2.21) (p. 96).
If INTERIOR is true, but  $\alpha_k < 0$  or  $\|s_k + \alpha_k p_k\|_M \geq \Delta$ , reset INTERIOR
    to false.
If INTERIOR is true, set
     $s_{k+1} = s_k + \alpha_k p_k$ .
    Otherwise, solve the tridiagonal trust-region subproblem (7.5.58) to ob-
        tain  $h_k$ .
End if
Set  $g_{k+1} = g_k + \alpha_k H p_k$ , and
     $v_{k+1} = M^{-1}g_{k+1}$ .
If INTERIOR is true, test for convergence using the residual  $\langle g_{k+1}, v_{k+1} \rangle$ 
    ( $\equiv \|g_{k+1}\|_{M^{-1}}$ ).
    Otherwise, test for convergence using the value  $\gamma_{k+1} |\langle e_{k+1}, h_k \rangle|$ .
End if
Set  $\beta_k = \langle g_{k+1}, v_{k+1} \rangle / \langle g_k, v_k \rangle$ , and
     $p_{k+1} = -v_{k+1} + \beta_k p_k$ .
If INTERIOR is false, recover  $s_k = Q_k h_k$  by rerunning the recurrences or obtaining
     $Q_k$  from backing store.

```

When recovering  $s_k = Q_k h_k$  by rerunning the recurrences, economies can be made by saving the  $\alpha_i$  and  $\beta_i$  during the first pass and reusing them during the second. A potentially bigger saving may be made if one is prepared to accept a slightly inferior value of the objective function. The idea is simply to save the value of  $q$  at each iteration. On convergence, one looks back through this list to find an iteration,  $\ell$  say, for which a required percentage of the best value was obtained, recompute  $h_\ell$ , and then accept  $s_\ell = Q_\ell h_\ell$  as the required estimate of the solution. If the required percentage occurs at an iteration before the boundary is encountered, both the final point before the boundary and the point delivered by the Steihaug–Toint method are suitable and available without the need for the second pass.

Notice that we have used the preconditioned conjugate gradient method (Algorithm 5.1.4 [p. 88]) to generate the Lanczos vectors. If the inner product  $\langle p_k, H p_k \rangle$  proves to be tiny, it is easy to continue using the preconditioned Lanczos method (Algorithm 5.2.4 [p. 104]) itself; the vectors

$$q_j = v_j / \sqrt{\langle g_j, v_j \rangle} \quad \text{and} \quad w_j = g_j / \sqrt{\langle g_j, v_j \rangle}$$

required to continue the Lanczos recurrence (5.2.50) (p. 104) are directly calculable from the preconditioned conjugate gradient method.

In view of Theorem 7.5.12, the irreducible tridiagonal trust-region subproblem (7.5.58) is, in theory, easier to solve than the general problem. This is so both because the Hessian is tridiagonal (and thus very inexpensive to factorize), and because the hard case cannot occur. We should be cautious here, because the so-called “almost” hard case—which occurs when  $g$  has only a tiny component in the range-space of  $H(\lambda^M)$ —may still happen, and the trust-region problem in this case is naturally ill-conditioned and thus likely to be difficult to solve. In the tridiagonal case, computing the extreme eigenvalues is straightforward, particularly if a sequence of related problems is to be solved. Thus, Algorithm 7.3.6, rather than Algorithm 7.3.4, is likely the most appropriate method to solve the subproblem.

A simple modification to Algorithm 7.3.6 is often useful here. As we are solving a sequence of problems in which  $T_k$  is merely  $T_{k-1}$  with an appended row and column, the solution value  $\lambda_{k-1}$  from the previous subproblem provides a natural starting approximation to  $\lambda_k$ . To check whether we can use this value, we need to see if it lies in the set  $\mathcal{L}_k$ , the set of values for which the Newton iteration will converge globally. For this to be so, we require that  $T_k + \lambda_{k-1} I_{k+1}$  be positive definite, and, if it is, that the resulting  $h_k(\lambda_{k-1})$ , as computed from

$$(T_k + \lambda I_{k+1})h_k(\lambda) = -\gamma_0 e_1,$$

lie outside the trust region, that is, that  $\|h_k(\lambda_{k-1})\|_2 \geq \Delta$ . As we will have already formed a factorization of  $T_{k-1} + \lambda_{k-1} I_k$  when solving the previous subproblem, it is trivial to obtain the factorization of  $T_k + \lambda_{k-1} I_{k+1}$ , and thus to determine if the latter is positive definite. If both of the required conditions are satisfied, we simply set  $\lambda = \lambda_{k-1}$ , and skip Steps 1 and 2 of Algorithm 7.3.6. The reader should be careful to remember that the  $s$  referred to in this algorithm is the  $h_k$  we are aiming for here.

We will not discuss methods for computing the leftmost Ritz value and, if required, its associated Ritz vector, but assure the reader that there are good methods for doing this. It is particularly important that we are solving a sequence of problems, as the leftmost value for one problem contains important information about its successor, and good methods take full advantage of this fact.

### Notes and References for Subsection 7.5.4

This section is based on Gould, Lucidi, Roma, and Toint (1999). Lemma 7.5.11 is due to Parlett (1980, Theorem 7.9.5). Computing the eigenvalues and vectors of a nested sequence of tridiagonal matrices is considered in detail by Parlett and Reid (1981). The Harwell Subroutine Library (2000) contains an implementation, **HSL\_VF05**, of the method considered here. A similar method for convex models was proposed by Lukšan (1996a). See also Hager (1999b).

The generalized Lanczos trust-region method is equally applicable when the problem involves affine constraints  $As = 0$ . The only alteration to Algorithm 7.5.2 required when solving (7.5.8) is to replace the preconditioning steps  $v_{k+1} = M^{-1}g_{k+1}$  by solutions of the linear system (5.4.5) (p. 110).

### 7.5.5 Computing the Eigenpoint

In Section 6.6, we saw that if we wish to ensure convergence to a second-order critical point, we need to be able to compute a suitable eigenpoint. Whenever negative curvature exists, such a point is obtained by approximately minimizing the given model along a significant direction of negative curvature, that is, one which gives at least a fixed percentage of the most negative curvature. Since the approximate minimization is straightforward once we have found such a direction, in this section we briefly focus on ways of computing a suitable direction of negative curvature.

If cost is of no concern, the simplest way is to compute the leftmost eigenvalue of the Hessian of the model,  $H$ , and then, if this value is negative, to compute a corresponding eigenvector. Means of doing this are discussed in the notes at the end of Section 2.2. Since such methods are typically more expensive than solving systems of linear equations, it is often worth first trying to compute the Cholesky or  $LDL^T$  factorization of  $H$ , and only computing the required eigenvector if the factorization fails or reveals that there is a negative eigenvalue—the factorization, if it succeeds, can be used to scale (precondition) the Cauchy step.

When the problem is large, such methods (or at least the eigencomputation part) are likely impractical for all but the most specially structured  $H$ . Fortunately, under these circumstances, it is most likely that we have resorted to one of the truncated conjugate gradient-like methods we considered in Sections 7.5.1 or 7.5.4 to compute an approximation to the model minimizer. As we noted in Section 5.2.5, these Krylov space-based methods may equally be used to compute approximate eigenvalues and, subsequently, eigenvectors of  $H$ . If we are using the Steihaug–Toint method, Algorithm 7.5.1, we may need to continue the computation of the Krylov space, using the

Lanczos method, if the step prematurely reaches the trust-region boundary. Fortunately, as we mentioned in the notes at the end of Section 5.2, since it is the extreme Ritz vectors that almost always converge first, we normally do not require a large number of Lanczos iterations to compute a good approximation to the required eigenvalue. Of course, the desired direction of negative curvature is then the corresponding approximate eigenvector and this requires that we either have saved the intermediate Krylov vectors or are willing to regenerate them in a second pass.

There is still one technical difficulty with such truncated conjugate gradient-like approaches, and this relates to the hard case. If the initial starting vector for the Lanczos method is  $M$ -orthogonal to the space of eigenvectors corresponding to the leftmost eigenvalue, this space will not be represented in the generated Krylov space. Thus, the leftmost Ritz value will converge to an interior (possibly even positive) eigenvalue of  $H$ , and the resulting eigenvector may not be uniformly related to the most negative curvature. The cure, if we know this has happened, is to restart the Lanczos method with a different initial vector, one that is ideally  $M$ -orthonormal to the previous Krylov space. In theory, this can only happen with probability zero, and in practice small rounding errors soon ensure that the required eigenspace has a nonnegligible contribution. Nevertheless, this is a theoretical disadvantage of the whole Krylov space approach, and one that can only be overcome for certain once the Krylov space has been expanded to cover  $\mathbb{R}^n$ . This phenomenon is essentially the same as that which we observed in Section 7.5.4 when the Lanczos tridiagonal  $T_k$  was reducible, and the cure is closely related to the conclusion of Theorem 7.5.17.

There is one special case where we can be certain that the initial space is insufficient, namely, when  $g = 0$ . Such a case would occur if, for example, the underlying minimization method were started from a saddle point, and we know that in this case the only way to escape is to compute a direction of negative curvature. If this occurs, we would simply suggest using the Lanczos method directly, picking any<sup>106</sup> initial vector. If we are solving a sequence of problems, the direction computed for one problem is often a very good starting vector for the next.

### 7.5.6 Eigenvalue-Based Approaches

The optimality conditions

$$(H + \lambda M)s = -g, \quad (7.5.65)$$

given by Theorem 7.4.1 for (7.5.1), are quite suggestive in another, rather punning, way. The scalar  $\lambda$  in this equation is the Lagrange multiplier corresponding to the trust-region constraint. Researchers in optimization frequently denote Lagrange multipliers by  $\lambda$ , while other disciplines reserve  $\lambda$  for eigenvalues. This is one case where we can

---

<sup>106</sup>In practice, it is best to pick a random vector, as certain “obvious” vectors, such as unit vectors or a vector of ones, are sometimes too “special” and lie in unfortunate invariant subspaces. Of course, there is no guarantee that any particular vector avoids this defect!

satisfy all factions at once, for (7.5.65) could equally have been written as

$$(H \ g) \begin{pmatrix} s \\ 1 \end{pmatrix} = -\lambda M s,$$

which looks suspiciously like the first block of a generalized eigenvalue problem—as indeed it is. Consider the eigenproblem

$$\begin{pmatrix} H & g \\ g^T & \theta \end{pmatrix} \begin{pmatrix} s \\ 1 \end{pmatrix} = (-\lambda) \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s \\ 1 \end{pmatrix}. \quad (7.5.66)$$

Here  $\theta$  is a given scalar,  $-\lambda$  is the eigenvalue we are seeking, and we have normalized its eigenvector so that its last component has the value 1.<sup>107</sup> Our aim is then to use the value  $\theta$  as a controlling parameter and to try to choose  $\theta$  so that a solution to (7.5.66) satisfies the remaining required optimality conditions, namely, that  $\lambda \geq 0$ ,  $H + \lambda M$  must be positive semidefinite, and  $\lambda(\|s\|_M - \Delta) = 0$ .

As the matrix on the right-hand side of (7.5.66) is symmetric and positive definite, while that on the left-hand side is symmetric, the eigenvalues must be real (see Section 2.2). The interlacing properties of eigenvalues of bordered matrices (see again Section 2.2) imply that we can ensure that  $H + \lambda M$  is positive semidefinite, by insisting that

$$\begin{pmatrix} H + \lambda M & g \\ g^T & \theta + \lambda \end{pmatrix}$$

is. Thus the generalized eigenvalue  $-\lambda$  we seek must be the leftmost; any other might allow  $H + \lambda M$  to be indefinite. Thus our remaining task is to satisfy the equation  $\lambda(\|s\|_M - \Delta) = 0$ .

Consider the function

$$\Phi(\lambda) = \lambda \Delta^2 - \langle g, s(\lambda) \rangle,$$

where we have expressed the dependence of  $s$  on  $\lambda$  implied by (7.5.65). Differentiating  $\Phi$  with respect to  $\lambda$  we obtain that

$$\Phi'(\lambda) = \Delta^2 - \langle g, \nabla_\lambda s(\lambda) \rangle \quad \text{and} \quad \Phi''(\lambda) = -\langle g, \nabla_{\lambda\lambda} s(\lambda) \rangle,$$

while differentiating (7.5.65) reveals that

$$\begin{aligned} (H + \lambda M) \nabla_\lambda s(\lambda) + M s(\lambda) &= 0 \quad \text{and} \\ (H + \lambda M) \nabla_{\lambda\lambda} s(\lambda) + 2M \nabla_\lambda s(\lambda) &= 0. \end{aligned} \quad (7.5.67)$$

Combining these equations with (7.5.65), we have that

$$\Phi'(\lambda) = \Delta^2 - \langle s(\lambda), M s(\lambda) \rangle \quad \text{and} \quad \Phi''(\lambda) = 2 \langle M s(\lambda), (H + \lambda M)^{-1} M s(\lambda) \rangle.$$

Thus, as  $\Phi''(\lambda) \geq 0$  for  $\lambda \geq -\lambda_1[H, M]$ ,  $\Phi$  is convex in this region, while  $\Phi'(\lambda) = 0$  when  $\|s(\lambda)\|_M = \Delta$ . Hence, if the solution lies on the boundary of the trust region,

---

<sup>107</sup>We could have constructed a nonsymmetric eigenproblem, but the advantages of symmetry are hard to ignore.

the value of  $\lambda$  we seek is the (global) minimizer of  $\Phi(\lambda)$  in the region  $\lambda \geq -\lambda_1[H, M]$ . We also note that there can only be an interior solution if  $\lambda_1[H, M] > 0$ , and that such a solution will occur if we encounter a  $\lambda \in (-\lambda_1[H, M], 0)$  for which  $\|s(\lambda)\|_M < \Delta$ , or alternatively for which  $\Phi'(\lambda) > 0$ ; once this situation has been identified, the required interior solution can be obtained by, for instance, applying the method of conjugate gradients to minimize  $q(s)$ . Thus in the remaining discussion, we shall assume that the solution lies on the trust-region boundary, and hence that we aim to minimize  $\Phi(\lambda)$ .

Recall that here we are not solving (7.5.65) to find  $s(\lambda)$ , but merely obtaining it as a by-product of the generalized eigenvalue calculation (7.5.66). In particular, we are not factorizing  $H + \lambda M$ , so obtaining the second derivative  $\Phi''(\lambda)$  is out of the question. Thus, if we intend to minimize  $\Phi$  by iteration, we are restricted to methods that only use function values and first derivatives. Fortunately, these are both directly available. Remember that  $\lambda$  is itself a function of the parameter  $\theta$ , and so we would actually minimize  $\Psi(\theta) \stackrel{\text{def}}{=} \Phi(\lambda(\theta))$ . The last block equation of (7.5.66) gives that

$$\langle g, s(\lambda(\theta)) \rangle + \theta = -\lambda(\theta), \quad (7.5.68)$$

so we may alternatively write

$$\Psi(\theta) = \theta + (1 + \Delta^2)\lambda(\theta). \quad (7.5.69)$$

It is easy to show that

$$\nabla_\theta \lambda(\theta) = -\frac{1}{1 + \|s(\lambda(\theta))\|_M^2}$$

by differentiating (7.5.68) with respect to  $\theta$ , substituting for  $\nabla_\lambda s(\lambda)$  from (7.5.67), and using (7.5.65). Hence, we also have that

$$\Psi'(\theta) = \frac{\|s(\lambda(\theta))\|_M^2 - \Delta^2}{1 + \|s(\lambda(\theta))\|_M^2}.$$

The standard way of minimizing  $\Psi$  is to use the minimizer of a suitable model as an estimate of the required minimizer of  $\Psi$ , with safeguards to prevent divergence if unsuitable values of  $\theta$  are generated. It is important that the model reflect the true nature of  $\Psi$ , which has a singularity at  $\lambda = -\lambda_1[H, M]$ . Provided that this is done, it is possible to derive rapidly convergent methods, at least in the easy case.

The observant reader will have already noticed that we have made a hidden assumption when deriving the above method. We said that we aim to find the leftmost generalized eigenvalue of (7.5.66) and that this will lead to the solution of (7.5.65) for some suitable  $\theta$ . However, it is possible that the last component of the eigenvector corresponding to this eigenvalue has the value zero, and thus we will not be able to renormalize the vector to make this last component 1. If this occurs, we have that

$$\begin{pmatrix} H & g \\ g^T & \theta \end{pmatrix} \begin{pmatrix} s \\ 0 \end{pmatrix} = (-\lambda_1) \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s \\ 0 \end{pmatrix}, \quad (7.5.70)$$

from which we immediately see that

$$Hs = (-\lambda_1)Ms \quad \text{and} \quad \langle g, s \rangle = 0.$$

But this is precisely the hard case:  $g$  is orthogonal to the set  $\mathcal{E}_1$  of eigenvectors corresponding to the leftmost generalized eigenvalue of  $(H, M)$ . At a first glance, this would appear to be a serious defect. But, in fact, this is not the case, for it also provides us with a suitable eigenvector in  $\mathcal{E}_1$  in the hard case.

Recall from Section 7.3.1.3 that the solution in the hard case may be made up from two components, one of which is the “generalized inverse” solution to

$$(H + \lambda_1 M)s_{\text{cri}} = -g$$

while the other is an eigenvector  $v \in \mathcal{E}_1$ . In the easy case,  $\lambda_1(\theta)$ , the leftmost eigenvalue of the pencil<sup>108</sup>

$$\left( \begin{pmatrix} H & g \\ g^T & \theta \end{pmatrix}, \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} \right),$$

is simple and strictly smaller than  $\lambda_1$ . To see this eliminate  $s$  from (7.5.66) to obtain the equation  $\lambda = \langle g, (H + \lambda M)^{-1}g \rangle - \theta$ , and notice that the two curves  $y = \lambda$  and  $y = \langle g, (H + \lambda M)^{-1}g \rangle - \theta$  intersect at a single point to the left of  $\lambda_1$ , the leftmost pole of  $\langle g, (H + \lambda M)^{-1}g \rangle$ . In the hard case, as we have seen,  $\lambda_1(\theta)$  has multiplicity at least as large as  $|\mathcal{E}_1|$  for all  $\theta$ , while the interlacing eigenvalue property shows that this multiplicity is at most  $|\mathcal{E}_1| + 1$ . As  $\theta$  increases, so does at least one of the remaining eigenvalues of the pencil. This implies that, for  $\theta$  larger than some critical value,  $-\lambda_1$  will no longer be the leftmost eigenvalue. What is the critical value? Quite simply

$$\theta = \theta_{\text{cri}} \stackrel{\text{def}}{=} -\lambda_1 + \langle g, s_{\text{cri}} \rangle.$$

For it is then easy to see that

$$\begin{pmatrix} H & g \\ g^T & \theta_{\text{cri}} \end{pmatrix} \begin{pmatrix} s_{\text{cri}} \\ 1 \end{pmatrix} = (-\lambda_1) \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} s_{\text{cri}} \\ 1 \end{pmatrix}.$$

As  $s_{\text{cri}}$  is orthogonal to  $\mathcal{E}_1$ , the eigenvector

$$\begin{pmatrix} s_{\text{cri}} \\ 1 \end{pmatrix} \tag{7.5.71}$$

and those of the form

$$\begin{pmatrix} s \\ 0 \end{pmatrix} \tag{7.5.72}$$

from (7.5.70) are orthogonal. Thus for this value, and this value alone, the multiplicity of  $\lambda_1(\theta)$  is  $|\mathcal{E}_1| + 1$ .

This then suggests how we might find the solution in the hard case. For some interval of values of  $\theta$  smaller than  $\theta_{\text{cri}}$ , the eigenvector of the second leftmost eigenvalue will converge to (7.5.71) as  $\theta$  increases to  $\theta_{\text{cri}}$ . So rather than just computing the leftmost eigenvalue and its vector, we should compute the two leftmost eigenvalues and related eigenvectors. The second leftmost will converge to (7.5.71) as  $\theta \rightarrow \theta_{\text{cri}}$ , while the limiting leftmost will be of the form (7.5.72), as given by (7.5.70).

---

<sup>108</sup>Let  $A$  and  $B$  be two  $n$  by  $n$  matrices. The set of all matrices of the form  $A - \lambda B$  with  $\lambda \in \mathbb{C}$  is said to be a pencil  $(A, B)$ .

### Notes and References for Subsection 7.5.6

This section is a distillation of the key ideas proposed by Rendl and Wolkowicz (1997), Sorensen (1997), and Santos and Sorensen (1995). Clearly, the essential ingredient is an efficient generalized eigensolution method. Rendl and Wolkowicz propose using the Lanczos method, while Santos and Sorensen prefer the iteratively restarted variant of this method. There are a number of alternatives to the merit function (7.5.69); Rendl and Wolkowicz suggest other alternatives based directly on the leftmost eigenvalues and interpret their method in terms of semidefinite programming. In any event, all of these methods appear to be extremely efficient and reliable. The exact methods used to adjust  $\theta$  and the relevant convergence properties are assessed in detail in the papers. For further background material see also Ben-Tal and Teboulle (1996), Rendl, Vanderbei, and Wolkowicz (1995), Stern and Wolkowicz (1994, 1995), and Fugger (1996). These ideas have been applied to the solution of ill-posed problems by Rojas (1998).

## 7.6 Projection-Based Approaches

Alternatively one may consider the trust-region subproblem (7.2.1) as a particular case of more general constrained optimization problems with a convex feasible set. Such problems are studied in Chapter 12, where we discuss algorithms based on taking advantage of the fact that, for some classes of feasible sets, the orthogonal projection of a vector onto the feasible set can be computed very efficiently. This is of course the case for the  $\ell_2$ -norm trust-region subproblem, where the feasible set is a scaled version of the unit sphere. We only mention here a simple iteration of the form

$$s_{k+1} = P_{\mathcal{B}} [s_k - \theta_k (H s_k + g)], \quad (7.6.1)$$

where the  $\theta_k$  are positive parameters and  $P_{\mathcal{B}}$  is the orthogonal projection operator onto the feasible set of (7.2.1). Note that  $H s_k + g = \nabla_x q(s_k)$ , which means that (7.6.1) can be interpreted as a “projected-gradient”-type method (again, see Chapter 12 for more detail) with stepsize  $\theta_k$ . Also note that (7.6.1) only requires one matrix-vector product per iteration, as is the case for other methods that we discussed above. It is easy to verify, using Theorem 7.2.1, that the solution of (7.2.1),  $s^M$ , is a fixed point of the recurrence (7.6.1). A whole class of methods for solving the trust-region subproblem may then be generated by considering suitable choices for the stepsizes  $\theta_k$ . Among them, it is interesting to single out the choice where, for each  $k$ ,

$$\theta_k = \frac{1}{\vartheta} \text{ with } \vartheta \geq \lambda_{\max}[H]. \quad (7.6.2)$$

With this particular choice, iteration (7.6.1) can be reinterpreted as an iteration of an algorithm that minimizes the difference of the two convex functions

$$q(s) = q_1(s) - q_2(s),$$

where

$$q_1(s) = \frac{1}{2}\vartheta\|s\|^2 + \langle g, s \rangle + \chi_{\mathcal{B}}(s) \text{ and } q_2(s) = \frac{1}{2}\langle s, (\vartheta I - H)s \rangle,$$

with  $\chi_{\mathcal{B}}(\cdot)$  being the indicator function<sup>109</sup> of the trust region  $\mathcal{B}$ .

The sequence  $\{s_k\}$  generated by this iteration can then be proved to converge to a first-order critical point of the trust-region subproblem, independently of the eigenstructure of  $H$  (and thus of a potential hard case). Although there is no formal guarantee that this point is a global minimizer, very often this seems to be the case in practice. Furthermore, one can show that the sequence  $\{q(s_k)\}$  is monotonically decreasing, which opens the possibility of truncating the iteration once sufficient reduction in the quadratic model has been obtained. Although efficient compared to the eigenvalue-based techniques of the preceding section, especially for the case where the minimizer lies on the boundary of the trust region or for the hard case, this method has the drawback of requiring an estimate of  $\lambda_{\max}[H]$ , which can be obtained either from the norm of  $H$  or from some other technique like Gershgorin bounds or a few steps of the Lanczos algorithm.

## Notes and References for Section 7.6

Iteration (7.6.1) with the choice (7.6.2) was introduced and studied by Pham Dinh and Le Thi (1998) under the name of the DC (difference of convex functions) algorithm. It arises as a specialized version of a much more general method (where  $q_1$  and  $q_2$  are allowed to be arbitrary, proper, lower semicontinuous convex functions) for the solution of the trust-region subproblem. A “restarted” variant of this algorithm also exists which can be shown to converge to the global solution of the trust-region subproblem, although at the potentially high cost of applying the local algorithm at most  $2n_- + 2$  times, where  $n_-$  is the number of distinct negative eigenvalues of  $H$ .

## 7.7 Norms that Reflect the Underlying Geometry

In Section 7.5, we noted that the methods we considered in Sections 7.3 and 7.4 might not, in general, be efficient. This lead us to consider more efficient methods whose aim was merely to find a useful approximation to the model minimizer. In this section we consider a second possibility, namely, that we may be able to find a norm for which the model minimizer is easy to obtain. After all, as we have already noted, it makes no difference in theory which norm we use within reason, and thus we may be wise to exploit this freedom.

### 7.7.1 The Ideal Trust Region

Suppose, for the time being, that  $H$  is nonsingular—we will relax this assumption in Section 7.7.4. We believe that the shape of an ideal trust region should reflect the geometry of the model and not give undeserved weight to certain directions. Indeed,

---

<sup>109</sup>An indicator function  $\chi_{\mathcal{B}}(s)$  for a set  $\mathcal{B}$  takes the value 0 if  $s \in \mathcal{B}$ , and  $\infty$  if  $s \notin \mathcal{B}$ .

perhaps the ideal trust region would be in the  $|H|$  norm, for which

$$\|s\|_{|H|}^2 = \langle s, |H|s \rangle, \quad (7.7.1)$$

and where the absolute value  $|H|$  is defined by (2.2.3) (p. 17). This norm reflects the proper scaling of the underlying problem—directions for which the model is changing fastest, and thus those for which the model may differ most from the true function, are restricted more than those directions for which the curvature is small. It has a further interesting property, namely, that a single matrix factorization (2.2.2) (p. 17) is needed to solve the problem. For, on writing

$$s_D = U^T s \quad \text{and} \quad g_D = U^T g,$$

and using the orthonormality of  $U$ , the solution of the trust-region subproblem may be expressed as  $s = Us_D$ , where  $s_D$  solves the *diagonal* trust-region subproblem

$$\underset{s_D \in \mathbb{R}^n}{\text{minimize}} \quad \langle g_D, s_D \rangle + \frac{1}{2} \langle s_D, \Lambda s_D \rangle \quad \text{subject to} \quad \langle s_D, |\Lambda|s_D \rangle \leq \Delta^2.$$

The diagonal trust-region subproblem is, as we shall shortly see, extremely inexpensive to solve. The major drawback of such an approach is, of course, the cost of the spectral factorization (2.2.2). For problems involving a large number of variables, this decomposition is likely out of the question.

### 7.7.2 The Absolute-Value Trust Region

With this in mind, we might instead consider the less expensive symmetric, indefinite factorization

$$H = PLBL^T P^T, \quad (7.7.2)$$

which we introduced in Section 4.3.4. We recall that here  $P$  is a permutation matrix,  $L$  is unit lower triangular, and  $B$  is block diagonal, with blocks of size at most 2. This factorization suggests that a natural choice for the trust-region norm is

$$\|s\|_M^2 = \langle s, Ms \rangle, \quad (7.7.3)$$

where

$$M = PL|B|L^T P^T. \quad (7.7.4)$$

Observe that  $|B|$  is simply computed by taking the absolute values of the 1 by 1 pivots and by forming an independent spectral decomposition of each of the 2 by 2 pivots and reversing the signs of any resulting negative eigenvalues. By analogy with the ideal method, writing

$$s_B = L^T P^T s \quad \text{and} \quad g_B = L^{-1} P^T g, \quad (7.7.5)$$

the solution of the trust-region subproblem may be expressed as  $s = PL^{-T}s_B$ , where  $s_B$  solves the *block diagonal* trust-region subproblem

$$\underset{s_B \in \mathbb{R}^n}{\text{minimize}} \quad \langle g_B, s_B \rangle + \frac{1}{2} \langle s_B, Bs_B \rangle \quad \text{subject to} \quad \langle s_B, |B|s_B \rangle \leq \Delta^2. \quad (7.7.6)$$

Once again, a single factorization suffices, but this time the factorization may be affordable even when  $n$  is large.

### 7.7.3 Solving Diagonal and Block Diagonal Trust-Region Subproblems

As the diagonal trust-region subproblem is a special (but not very special) case of the block diagonal case, here we shall concentrate on the latter. One could simply apply a standard method, like Algorithm 7.3.4 or more especially Algorithm 7.3.6, to (7.7.6), but we prefer not to do this, as this would, to some extent, ignore the structure in hand.

As  $B$  and  $|B|$  share eigenvectors, we may write

$$B = Q\Theta Q^T \quad \text{and} \quad |B| = Q|\Theta|Q^T,$$

where each column of  $Q$  is nonzero in at most two positions, with entries corresponding to the eigenvectors of the diagonal blocks, and where the entries of the diagonal matrix  $\Theta$  are the corresponding eigenvalues. On defining

$$s_s = |\Theta|^{\frac{1}{2}}Q^T s_B \quad \text{and} \quad g_s = |\Theta|^{-\frac{1}{2}}Q^T g_B,$$

we may solve (7.7.6) by finding  $s_s$  to

$$\underset{s_s \in \mathbb{R}^n}{\text{minimize}} \quad \langle g_s, s_s \rangle + \frac{1}{2}\langle s_s, Ds_s \rangle \quad \text{subject to} \quad \|s_s\|_2 \leq \Delta, \quad (7.7.7)$$

and then recover  $s_B = Q|\Theta|^{-\frac{1}{2}}s_s$ . Significantly, the matrix  $D \equiv |\Theta|^{-\frac{1}{2}}\Theta|\Theta|^{-\frac{1}{2}}$  is diagonal with entries  $\pm 1$ . The required solution must then satisfy

$$(D + \lambda I)s_s = -g_s, \quad (7.7.8)$$

where the nonnegative Lagrange multiplier  $\lambda$  is sufficiently large to ensure that  $D + \lambda I$  is positive semidefinite, and is zero if  $s_s$  lies within the trust region  $\|s_s\|_2 \leq \Delta$ .

There are two cases to consider. Firstly, if  $D = I$ , the solution to (7.7.8) is

$$s_s = -\frac{1}{1+\lambda}g_s.$$

If  $\|g_s\|_2 < \Delta$ , the solution to (7.7.7) is given by  $s_s = -g_s$  and  $\lambda = 0$ . This corresponds to the unconstrained minimizer of the model lying interior to the trust region. If, on the other hand,  $\|g_s\|_2 \geq \Delta$ , the solution to (7.7.7) is obtained by finding the value of  $\lambda \geq 0$  for which

$$\frac{1}{(1+\lambda)}\|g_s\|_2 = \Delta.$$

This is a linear equation in  $\lambda$  and thus the solution is trivial to obtain; the required  $s_s$  is

$$s_s = -\frac{\Delta}{\|g_s\|_2}g_s.$$

This corresponds to the case where the model is convex, but the trust region excludes the unconstrained minimizer of the model. Notice also that, in this case, a reduction in the trust-region radius following an unsuccessful step merely reduces the length of

the step in the direction  $-g_s$ . Such a strategy is identical in its effect (if not in its motivation) to a backtracking linesearch along the quasi-Newton direction  $-H^{-1}g$ , and thus there is a strong similarity between trust-region and linesearch methods with this choice of trust region.

Secondly, if  $H$  has negative eigenvalues,  $D$  will have some diagonal entries of  $-1$ . Suppose  $P_s$  is a permutation matrix which arranges that all the positive diagonals (+1) of  $D$  precede its negative diagonals (-1). Then it is easy to show that

$$s_s = -\frac{1}{\lambda^2 - 1} P_s^T \begin{pmatrix} (\lambda - 1)I & 0 \\ 0 & (\lambda + 1)I \end{pmatrix} P_s g_s. \quad (7.7.9)$$

As  $H$  is indefinite, the solution must lie on the trust-region boundary. Thus, we may obtain  $\lambda$  as a root of the quartic equation

$$\left\langle P_s g_s, \begin{pmatrix} (\lambda - 1)^2 I & 0 \\ 0 & (\lambda + 1)^2 I \end{pmatrix} P_s g_s \right\rangle = (\lambda^2 - 1)^2 \Delta^2,$$

while applying Corollary 7.2.2 to the problem (7.7.7) shows that the root we seek is the one larger than 1. Although in principle this root may be found explicitly by Ferrari's method (see, for instance, Turnbull, 1939, and Salzer, 1960), Newton's method is equally suitable here. A slight complication may occur when all of the components of  $P_s g_s$  corresponding to the negative diagonals of  $D$  are zero. Then (7.7.9) yields

$$s_s = -\frac{1}{\lambda + 1} P_s^T \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_s g_s,$$

and it may be that there is no root larger than 1 of the resulting feasibility equation

$$\left\langle P_s g_s, \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_s g_s \right\rangle = (\lambda + 1)^2 \Delta^2.$$

This corresponds to the hard case, and, as usual, the solution includes a contribution from a suitable eigenvector. In our case, it is of the form

$$s_s(\alpha) = -\frac{1}{2} P_s^T \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} P_s g_s + \alpha P_s^T \begin{pmatrix} 0 \\ u \end{pmatrix},$$

where  $u$  is any nonzero vector and  $\alpha$  is chosen as a root of the quadratic equation  $\langle s_s(\alpha), s_s(\alpha) \rangle = \Delta^2$ .

#### 7.7.4 Coping with Singularity

Clearly, it is important to deal with any matrix  $H$  including those that are, or are close to being, singular. A simple solution is to compute the factorization (7.7.4) and to replace each eigenvalue  $\theta$  of the block diagonal  $B$  with the value

$$\gamma = \begin{cases} \theta & \text{if } \theta \geq \delta \text{ or} \\ \delta & \text{otherwise} \end{cases} \quad (7.7.10)$$

for some small  $\delta > 0$ . An alternative, which is closer in spirit to the absolute value perturbation, is to replace each eigenvalue by

$$\gamma = \begin{cases} \theta & \text{if } \theta \geq \delta \text{ or} \\ -\theta & \text{if } \theta \leq -\delta \text{ or} \\ \delta & \text{otherwise.} \end{cases} \quad (7.7.11)$$

In any event, this does not significantly affect our previous discussion. For, if we let  $C$  denote the (possibly) modified block diagonal matrix  $B$ , we now use the trust-region norm (7.7.3) with  $M$  defined as

$$M = PLCL^T P^T. \quad (7.7.12)$$

The decomposition (7.7.12) is known as the *modified absolute-value* factorization. If we make the change of variables (7.7.5), we must solve the block diagonal trust-region subproblem

$$\underset{s_B \in \mathbb{R}^n}{\text{minimize}} \quad \langle g_B, s_B \rangle + \frac{1}{2} \langle s_B, B s_B \rangle \quad \text{subject to} \quad \langle s_B, C s_B \rangle \leq \Delta^2. \quad (7.7.13)$$

It is of little consequence that  $BC^{-1}$  no longer necessarily has eigenvalues  $\pm 1$ , for, as we shall now see, solving the problem (7.7.13) is also straightforward.

As before,  $B$  and  $C$  share eigenvectors. We may thus write

$$B = Q\Theta Q^T \quad \text{and} \quad C = Q\Gamma Q^T,$$

where  $Q$  is as before, and the entries of the diagonal matrices  $\Theta$  and  $\Gamma$  are, respectively, the values  $\theta$  and  $\gamma$  considered in (7.7.10) or (7.7.11). Using the transformation

$$s_S = \Gamma^{\frac{1}{2}} Q^T s_B \quad \text{and} \quad g_S = \Gamma^{-\frac{1}{2}} Q^T g_B,$$

we may recover the solution to (7.7.13) from  $s_B = Q\Gamma^{-\frac{1}{2}} s_S$ , where  $s_S$  is found to

$$\underset{s_S \in \mathbb{R}^n}{\text{minimize}} \quad q_S(s_S) \equiv \langle g_S, s_S \rangle + \frac{1}{2} \langle s_S, D s_S \rangle \quad \text{subject to} \quad \|s_S\|_2 \leq \Delta, \quad (7.7.14)$$

and where  $D \equiv \Gamma^{-\frac{1}{2}} \Theta \Gamma^{-\frac{1}{2}}$  is diagonal. Once again, one could simply apply Algorithms 7.3.4 or 7.3.6 to this problem, but this ignores the facts that the diagonal systems involved are trivial to solve and that the leftmost eigenvalue of  $D$  and a corresponding eigenvector are trivial to obtain. We therefore prefer the following simplification.

If  $D$  merely has entries  $\pm 1$ , the procedure outlined in Section 7.7.3 is appropriate. So now suppose that  $D$  has a more complicated distribution of values. Then we may apply Algorithm 7.7.1.

**Algorithm 7.7.1: Newton iteration to solve (7.7.14)**

Let  $\epsilon \in (0, 1)$ .

**Step 1.** If  $D$  is positive definite, set  $\lambda = 0$  and  $s_s = -D^{-1}g_s$ .

**Step 1a.** If  $\|s_s\|_2 \leq \Delta$ , stop.

**Step 2.** Otherwise, compute the leftmost eigenvalue,  $\theta$ , of  $D$ , set  $\lambda = -\theta$ , and define  $g_s^n$  so that

$$(g_s^n)_i = \begin{cases} (g_s)_i & \text{if } (D)_{ii} + \lambda = 0 \\ 0 & \text{otherwise.} \end{cases}$$

**Step 2a.** If  $g_s^n = 0$ , set  $s_s = -(D + \lambda I)^+ g_s$ .

(i) If  $\|s_s\|_2 \leq \Delta$ , compute an eigenvector  $u$  corresponding to  $\theta$ , find the root  $\alpha$  of the equation  $\|s_s + \alpha u\|_2 = \Delta$  which makes the model  $q_s(s_s + \alpha u)$  smallest, replace  $s_s$  by  $s_s + \alpha u$ , and stop.

**Step 2b.** Otherwise, replace  $\lambda$  by  $\lambda + \|g_s^n\|_2 / \Delta$ , and set  $s_s = -(D + \lambda I)^{-1} g_s$ .

**Step 3.** If

$$|\|s_s\|_2 - \Delta| \leq \epsilon \Delta,$$

stop.

**Step 4.** Replace  $\lambda$  by  $\lambda + \left( \frac{\|s_s\|_2 - \Delta}{\Delta} \right) \left( \frac{\|s_s\|_2^2}{\langle s_s, (D + \lambda I)^+ s_s \rangle} \right)$ .

**Step 5.** Set  $s_s = -(D + \lambda I)^+ g_s$  and go to Step 3.

The iteration in Steps 3 to 5 is simply Newton's method to find the appropriate root of the secular equation

$$\frac{1}{\|-(D + \lambda I)^+ g_s\|_2} = \frac{1}{\Delta}.$$

Step 1 caters to the case where the model is strictly convex, while Step 2 is for the more general case where the solution must lie on the trust-region boundary. The precaution in Step 1a is simply to detect the solution when it lies interior to the trust region, while that in Step 2a(i) is to compute the solution in the hard case. The iteration is globally Q-linearly and asymptotically Q-quadratically convergent from the starting values given in Steps 1 and 2. We stress that, while this algorithm is appropriate even if  $D$  is simply a diagonal matrix with entries  $\pm 1$ , the procedure outlined in Section 7.7.3 is more appropriate in this case.

It remains to show that the norms defined by the modified absolute-value factorization (7.7.12) are uniformly equivalent (see AN.1), and thus are suitable within a trust-region method. Thus we need to show that there are constants  $0 < \gamma_1 < \gamma_2$ ,

independent of the iteration, for which

$$\gamma_1 \|s\|_2^2 \leq \langle s, Ms \rangle \leq \gamma_2 \|s\|_2^2.$$

Equivalently, we need to show that the smallest and largest eigenvalues,  $\lambda_{\min}(M)$  and  $\lambda_{\max}(M)$ , of  $M$  are bounded, and bounded away from zero.

To see this, we first observe that both (7.7.10) and (7.7.11) satisfy the bounds

$$\delta \leq \|B\| = \max [\delta, \|D\|] \leq \max [\delta, \|H\| \|LL^T)^{-1}\|].$$

It is then straightforward to show that

$$\lambda_{\min}(LL^T)\lambda_{\min}(C) \leq \lambda_{\min}(M) \leq \lambda_{\max}(M) \leq \lambda_{\max}(LL^T)\lambda_{\max}(C) \quad (7.7.15)$$

and hence that

$$\delta\lambda_{\min}(LL^T) \leq \lambda_{\min}(M) \leq \lambda_{\max}(M) \leq \lambda_{\max}(LL^T) \max [\delta, \|H\| \|LL^T)^{-1}\|].$$

But, if the largest entry in  $L$  is bounded by some  $\beta$ , it is again straightforward to bound

$$1 \leq \lambda_{\max}(LL^T) \leq n + \frac{1}{2}n(n-1)\beta^2 \quad \text{and} \quad (1+\beta)^{2-2n} \leq \lambda_{\min}(LL^T) \leq 1. \quad (7.7.16)$$

Thus so long as  $L$  and  $H$  are bounded, the norms defined by the modified absolute-value factorization (7.7.12) are uniformly equivalent. The matrix  $H$  will be bounded if, for instance, a Newton (second-order Taylor approximation) model is used. But now we see the importance of using a factorization that bounds the growth in the elements of  $L$ . Fortunately many, but not all, symmetric indefinite factorizations provide such bounds.

## Notes and References for Section 7.7

The material in this section is due to Gould and Nocedal (1998). The absolute-value factorization (2.2.3) (p. 17) was originally proposed by Greenstadt (1967) in conjunction with line-search methods for unconstrained minimization. Gill, Murray, Ponceleón, and Saunders (1992) proposed the modified factorization (7.7.4) as a preconditioner for iterative methods, while Cheng and Higham (1998) suggest it as an alternative to the modified Cholesky factorizations of Gill and Murray (1974), Gill, Murray, and Wright (1981), and Schnabel and Eskow (1991, 1999) within linesearch-based methods. The modification (7.7.10) is due to Cheng and Higham (1998), while (7.7.11) is preferred by Gould and Nocedal (1998). Others have used the factorization (7.7.2) to define trust-region norms. Goldfarb (1980) suggested using (7.7.3), but with (7.7.4) replaced by

$$M = PLL^T P^T. \quad (7.7.17)$$

Following the change of variables (7.7.5), the resulting block diagonal trust region is then of the form

$$\underset{s_B \in \mathbb{R}^n}{\text{minimize}} \quad \langle g_B, s_B \rangle + \frac{1}{2} \langle s_B, B s_B \rangle \quad \text{subject to} \quad \|s_B\| \leq \Delta$$

and its solution is again straightforward to obtain. This idea has been further explored by Xu and Zhang (1999). We believe, however, that using (7.7.17) rather than (7.7.4) does not reflect the proper scaling of the underlying problem. Indeed, if  $H$  were a diagonal matrix, (7.7.3) would remain as the  $\ell_2$  norm regardless of how ill-conditioned  $H$  might be.

The important bounds (7.7.15) and (7.7.16) are due to Higham (1995). Ashcraft, Grimes, and Lewis (1995) show that the original method of Bunch and Parlett (1971) and that of Fletcher (1976) both generate bounded  $L$ , as do the sparse methods of Duff and Reid (1983, 1996). However, the more popular Bunch and Kaufman (1977) method and the block version implemented in LAPACK may not and thus must be viewed as untrustworthy for this application.

There are some potential difficulties with the whole approach. For some large problems, the decomposition (7.7.2) may prove to be too expensive, particularly if significant fill-in occurs in the factors. The attendees at the 1981 NATO Advanced Research Institute on Nonlinear Optimization (see Powell, 1982, contributions 1.31–1.35) had much to say about Goldfarb's (1980) proposal, and the comments made there are equally appropriate here. In particular, there was some concern that the distortion induced by (7.7.3) and (7.7.17) may be substantial. While it is clear that (7.7.12) may not be as ideal as (7.7.1), the latter is out of the question for most large-scale problems, whereas (7.7.12) is practical, and often useful, for many of them. In particular Gould and Nocedal (1998) found that (7.7.12) was particularly effective for badly scaled problems. Concerns were also expressed in 1981 that changes in the pivot ordering during the factorization of a sequence of problems might make it difficult to derive effective methods for adjusting the trust-region radius. While Gould and Nocedal (1998) observed occasions where pivot-order changes drastically altered the geometry, and while this sometimes required a large number of wasted iterations in which the trust-region radius is reduced, for the vast majority of iterations the usual, naive trust-region management appeared to be satisfactory. A code, **HSL\_VF06**, which implements this method, is available in the Harwell Subroutine Library (2000).

## 7.8 The $\ell_\infty$ -Norm Problem

To find the model minimizer of the  $\ell_2$ -norm trust-region problem, we now turn, briefly, to the other trust-region norm that commonly occurs in practice, the  $\ell_\infty$  norm. This norm is appealing for a number of reasons. Firstly, restricting  $\|s\|_\infty \leq \Delta$  merely means that the simple bounds

$$-\Delta \leq s_i \leq \Delta \quad (7.8.1)$$

must be satisfied componentwise. In particular, it is very easy to check a point for feasibility. Moreover, if, as is extremely common, the minimization problem itself involves bounds  $l \leq x \leq u$ , the trust-region solution ensures feasibility by requiring that  $s$  lie in the intersection of the trust region and the problem bounds  $l - x \leq s \leq u - x$ . That is, the components of  $s$  are simply required to satisfy

$$\max[l_i - x_i, -\Delta] \leq s_i \leq \min[u_i - x_i, \Delta].$$

The geometry of the “box” shapes of the  $\ell_\infty$  norm and of the simple bounds may then be simply exploited, as the intersection of two or more right-oriented “boxes” is a “box”.

There are, unfortunately, two significant reasons to prefer the  $\ell_2$  norm. The first is that there are polynomial time algorithms for solving the  $\ell_2$ -norm problem (see Section 7.3), while, at least when  $H$  is indefinite, the  $\ell_\infty$ -norm problem lies in NP hard, a class of problems for which it is believed that it is unlikely that there are polynomial algorithms. That this might be true is reflected by the fact that each variable may be in one of three possible states (on its lower bound, between bounds, or on its upper bound), giving  $3^n$  possible solution states. When  $H$  is nonconvex, there may be isolated local minimizers in at least  $2^n$  of these (for example, local minimizers of  $-\langle x, x \rangle$  subject to  $-\Delta \leq x \leq \Delta$  occur at each of the  $2^n$  vertices of the trust region) and each minimizer is unaware of the presence of the others. By contrast, the minimizer of the  $\ell_2$ -norm problem lies in one of only two possible states (interior to the trust region or on the smooth trust-region boundary), and this suggests one reason why the  $\ell_2$ -norm problem might be easier from a complexity-theoretical viewpoint. Worse still, if we narrow our sights, and merely aim to compute a local minimizer, this seemingly easier problem is still NP hard, as is the problem of determining whether a known first-order critical point is actually a local minimizer. When  $H$  is positive semidefinite, there are a number of polynomial time interior-point methods for solving the  $\ell_\infty$ -norm problem, but requiring that  $H$  be positive semidefinite for any model problem we encounter is tantamount to assuming that the underlying nonlinear problem is convex.

The second reason why the  $\ell_\infty$  norm may be less convenient is related to the basic method for solving large-scale quadratic minimization problems, the conjugate gradient method (see Section 5.1). One obvious way to solve the problem is to apply conjugate gradients, starting from the initial point  $s = 0$ . We saw in Theorem 7.5.1 that, if we use a (scaled)  $\ell_2$  norm, the norms of the solution estimates increase. Thus, if an estimate  $s_k$  is generated outside the trust region, we can be quite confident that the solution to the trust-region problem must lie on the trust-region boundary. This is exploited in the methods described in Sections 7.5.1–7.5.4. Unfortunately, this property does not hold when the  $\ell_\infty$  norm is used. That is, it can happen that the conjugate gradient estimate  $s_k$  lies outside the trust region, but the solution lies inside—the conjugate gradient path will reenter the trust region at a later stage. See Figure 7.8.1 for an example.

Despite these shortcomings, the  $\ell_\infty$  norm remains attractive, particularly since the theory developed in Chapter 6 requires that in practice we need go no further than the Cauchy point, and this at least is just as simple to compute in the  $\ell_\infty$  norm. More importantly, there are a number of potentially inefficient (in a complexity sense) but nevertheless practically effective methods for finding a local first-order critical point for the problem.

The earlier of these are active, or, as they are now perhaps more correctly known, *working set methods*, and may best be thought of as a form of intelligent enumeration. The basic idea is simply to try to predict which of the  $2n$  bound constraints (7.8.1) are active at a local minimizer of

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && q(s) \equiv \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle \\ & \text{subject to} && \|s\|_\infty \leq \Delta. \end{aligned} \tag{7.8.2}$$

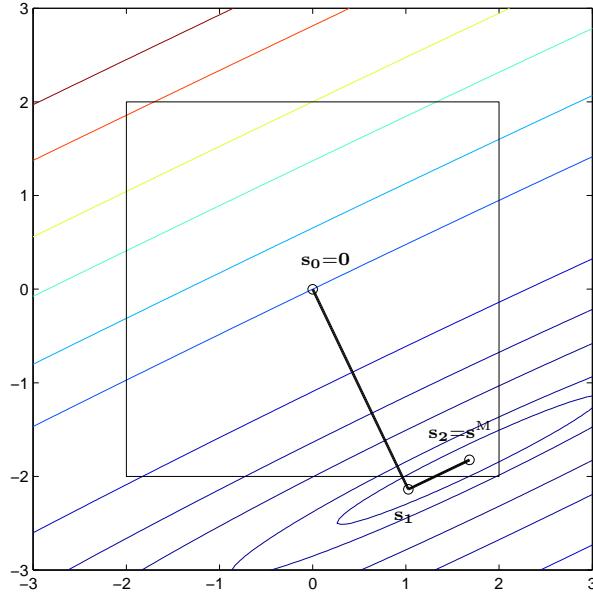


Figure 7.8.1: The conjugate gradient path starting from the origin. Notice that the first iterate  $s_1$  lies outside the trust region  $\|s\|_\infty \leq 2$ , while the solution  $s_2$  lies inside.

If this set were known in advance, the relevant variables could be fixed at their bounds and the *unconstrained* minimizer of  $q$  with respect to the remaining variables sought, using either a direct method (see Section 4.3) or an iterative one (see Section 5.1). The predicted set of active bounds is known as the *working set*. The art is in knowing which constraints to include in the working set.

The simplest method works as follows. Let  $i$  be an integer vector whose  $j$ th component has values 0 or  $\pm 1$ , and let  $\mathcal{W}(i) = \{j \mid i_j = 1\} \cup \{-j \mid i_j = -1\}$ . The vector  $i$  is intended to be an indicator vector for the status of each variable: a variable  $s_j$  will be fixed at its lower bound,  $-\Delta$ , if  $i_j = -1$ , and fixed at its upper bound,  $\Delta$ , if  $i_j = 1$ . Similarly,  $\mathcal{W}(i)$  will be the working set indicated by  $i$ , a negative element  $-j$  requires that  $s_j = -\Delta$ , and a positive one  $j$  requires that  $s_j = \Delta$ .

To start, pick an arbitrary  $s$  within the trust region, define the indicator vector  $i$  by the status of each of the components of  $s$ , and let  $\mathcal{W} \stackrel{\text{def}}{=} \mathcal{W}(i)$ . At each iteration, the variables in  $\mathcal{W}$  are temporarily frozen on their bounds, and we attempt to minimize  $q$  with respect to the remaining free variables. That is, on writing<sup>110</sup>

$$s = \begin{pmatrix} s^F \\ s^B \end{pmatrix}, \quad g = \begin{pmatrix} g^F \\ g^B \end{pmatrix}, \quad \text{and} \quad H = \begin{pmatrix} H^{FF} & H^{FB} \\ H^{FB^T} & H^{BB} \end{pmatrix},$$

where  $s^B$  are the variables that are frozen at their bounds  $\pm\Delta$  by  $\mathcal{W}$  and  $s^F$  are the

---

<sup>110</sup>Ignoring any necessary permutations.

remaining free ones, we aim to

$$\underset{s^F \in \mathbb{R}^n}{\text{minimize}} \quad q^F(s^F) \equiv \langle g^F + H^{FB}s^B, s^F \rangle + \frac{1}{2}\langle s^F, H^{FF}s^F \rangle. \quad (7.8.3)$$

There are three possibilities. Firstly,  $q^F(s^F)$  may be unbounded from below. In this case, the required minimizer cannot be defined by  $\mathcal{W}$  and there is a direction  $d^F$  such that  $q^F(s^F + \alpha d^F)$  is a decreasing function for all  $\alpha \geq 0$ . Moving along the line  $s^F + \alpha d^F$ , we must eventually cross the boundary of the trust region. The iteration is concluded by setting  $s$  to this boundary point and picking  $i$  and  $\mathcal{W}$  to reflect the status of the components of  $s$ . The second case is when (7.8.3) has a finite minimizer  $s^F + d^F$ , but  $s^F + d^F$  beyond the trust-region boundary. In this case, once again,  $q^F$  decreases along the line  $s^F + \alpha d^F$ , and the iteration terminates at the point  $s$  at which the line crosses the boundary with the appropriate  $i$  and  $\mathcal{W}$ . Finally, it may happen that (7.8.3) has a finite minimizer  $s^F + d^F$  within the trust region. Such a point is known as a *feasible solution to the equality problem* (FSEP). In this case, we set  $s$  to this value but must now check to see if  $s$  is a first-order critical point or if we can progress further—clearly  $s$  is the minimizer of  $q$  defined by the active set at  $s$ . To do this, we simply examine the signs of the components of  $g + Hs$  for indices  $j \notin \mathcal{W}$ . Those on  $\mathcal{W}$  are, by construction, zero. The first-order optimality conditions (3.2.4) (p. 40) applied to (7.8.2) are simply that

$$(g + Hs)_i \geq 0 \text{ if } s_i = -\Delta \text{ and } (g + Hs)_i \leq 0 \text{ if } s_i = \Delta, \quad (7.8.4)$$

and the iteration is terminated if these conditions hold. Otherwise, the components of  $i$  of any (or all) of the indices which violate these conditions can be reset to 0, and  $\mathcal{W}$  adjusted appropriately before starting the next iteration. Since the third case discussed above can only happen a finite number (at most  $3^n$ ) of times, and since there can only be a finite number (at most  $n$ ) of iterations between occurrences of this third case, the algorithm is finite (but, of course, it might possibly require an exponential number of steps).

The above description is not complete since we have not described—nor do we intend to—the algebraic manipulation required to compute the step at each iteration. Suffice to say, this may be accomplished using either direct matrix factorization, in which economies are possible by updating rather than recomputing the factorizations to account for the gradual changes in the working set, or by iterative methods like the conjugate gradient method.

Such an algorithm allows us to discard the indices of as many variables whose components violate (7.8.4) as we like from the working set at an FSEP. Thus rapid changes in the working set are possible. Unfortunately, iterations that do not lead to FSEPs may, and usually do, result in very gradual increases in the working set. While this is not necessarily a disadvantage—the (normally dominant) linear algebraic costs following minor changes to the working set are often considerably smaller than those that occur when the working set is radically altered—it may still be preferable to use methods that allow rapid changes at every iteration.

An appealing method of this sort is based on the notion of the projected-gradient path. We shall discuss this in detail, in more generality, in Section 12.1.3, but here we shall briefly explain the idea. The projected-gradient path for our problem from a given point  $s$  is the arc  $P(s, t)$ , which is given componentwise for all  $t \geq 0$  as

$$P(s, t)_i = \text{mid}(-\Delta, s_i - t[\nabla_s q(s)]_i, \Delta)$$

for  $i = 1, \dots, n$ , where  $\text{mid}(a, b, c)$  denotes the median value of  $a$ ,  $b$ , and  $c$ . This path has the crucial property that a given  $s$  is critical if and only if  $P(s, t) = s$  for all  $t \geq 0$  (see Theorem 12.1.2). Thus, if  $s$  is not critical, the path leads away from  $s$ . It is also possible to show that, in this case,  $q(P(s, t))$  decreases for  $0 < t \leq t_1$  for some  $t_1 > 0$ . So long as a “significant” step  $t$  is taken, the very simple algorithm resulting from replacing  $s$  by the improvement  $P(s, t)$  can be shown to converge to a local first-order critical point. “Significant” here could mean finding a local minimizer of the piecewise quadratic univariate function  $q(P(s, t))$ , or perhaps performing an Armijo-type backtracking search from the end of the path.

The reader will probably realize that such an algorithm is, by itself, unlikely to be very efficient since, if  $\Delta$  is very large, the method is essentially the method of steepest descent. More practical methods interweave projected-gradient path iterations with “fast” iterations aimed at improving the rate of convergence. Often the active set at the chosen point on the projected-gradient path is used as the working set, and a few iterations of the working set method described earlier in this section is used from this point. The way in which the two types of iterations are mixed is important and defines the variety of competing proposals that have been made. Methods of this type have a further advantage in that the Cauchy point for the underlying problem for which (7.8.2) is a subproblem is often obtained as the first iterate, since the Cauchy arc and the projected-gradient path from  $s = 0$  are the same for (7.8.2). Thus convergence of the overall method is ensured since any additional iterates satisfy the assumptions of Theorem 6.3.5 (p. 132).

Finally, we must mention a further possibility that we will consider in more detail, not to mention generality, in Chapter 13. This is to solve (7.8.2) by an interior-point/barrier function method. While this may seem an awfully big hammer to crack such a simple nut—the underlying method would appear to increase the nonlinearity of the subproblems to be solved—there is a growing appreciation that such methods are relatively insensitive to the dimension of the problem, and thus, at least for large problems, they may be the best overall choice if an accurate local first-order critical point is required. Whether intermediate iterates are useful in the trust-region context is, as yet, less clear.

## Notes and References for Section 7.8

Murty and Kabadi (1987) and Vavasis (1992b) showed that the nonconvex  $\ell_\infty$ -norm trust-region problem is NP hard. Effective polynomial time interior-point methods for the convex

problem have been suggested by a number of authors including Ye and Tse (1989). Interestingly, Ye (1997) shows that, although the optimal solution may not be found, in general, in polynomial time, there is a polynomial time algorithm for finding an estimate  $s$  of the solution  $s^M$  for which

$$q(s) - q(s^M) \leq \frac{4}{7} (q(s^U) - q(s^M)), \quad (7.8.5)$$

where  $s^U$  is the global maximizer of  $q$  within the trust region. Unfortunately, it is also known that in general the constant  $\frac{4}{7}$  in (7.8.5) cannot be reduced very much without losing polynomiality—Bellare and Rogaway (1995) indicate that lowering the value to  $\frac{1}{4}$  is already NP hard.

Active/working set methods for solving (7.8.2) have been proposed by Polyak (1969), Fletcher and Jackson (1974), Gill and Murray (1976), O’Leary (1980), Coleman and Hulbert (1989), Júdice and Pires (1989), and Yang and Tolle (1991). The projected-gradient path method is essentially due to Bertsekas (1982b) and Dembo and Tulowitzki (1983), while Moré and Toraldo (1991), Friedlander and Martínez (1994), Barlow and Toraldo (1995), Friedlander, Martínez, and Raydan (1995), McKenna, Mesirov, and Zenios (1995), Dostál (1997), and Diniz-Ehrhardt, Gomes-Ruggiero, and Santos (1998) are responsible for the current state of the art (see also the notes to Sections 12.1.4 and 12.2.2). Ye (1989), Han, Pardalos, and Ye (1992), Conn, Gould, and Toint (1994), and Conn, Gould, Orban, and Toint (2000) have proposed special-purpose interior-point methods for bound-constrained quadratic programs. See also Chapter 13, as well as Coleman and Hulbert (1993), Li and Swetits (1993), Li (1996), Coleman and Li (1996b), and Coleman and Liu (1999).

# Chapter 8

---

## Further Convergence Theory Issues

---

Having considered the basic trust-region algorithm and its convergence theory in Chapter 6, we now turn to theoretical extensions of our basic framework. Although each of these extensions is interesting, not all of them are crucial for understanding the trust-region paradigm. The reader may thus skip any or all sections of this chapter, excepting the first, without losing the continuity of our exposition.

### 8.1 Convergence of General Measures

#### 8.1.1 First-Order and Criticality Measures

So far we have considered the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x),$$

for which we developed a trust-region algorithm (and variants) that converges to first-order critical points. This algorithm produces a sequence of iterates  $\{x_k\}$  and, for each of these iterates, we have measured how closely  $x_k$  satisfies the property that we seek (a zero gradient) by computing  $\|g_k\|$ . And our theory then shows that, so long as AA.1 and other assorted assumptions hold, we have that

$$\lim_{k \rightarrow \infty} \|g_k\| = 0;$$

that is, the desired first-order criticality property is asymptotically satisfied. The purpose of this section is to show that the same mechanism can be used to prove the convergence to zero of other measures of the “desirability” of the iterate  $x_k$ . We shall exploit this generality in later sections of our book.

In order to formalize this concept, we first assume that we wish our trust-region algorithm to find an iterate  $x_k \in \mathbb{R}^n$  for which some desirable property holds asymptotically. We write this in the form

$$\lim_{k \rightarrow \infty} \pi(k, x_k) = 0. \quad (8.1.1)$$

For instance, in the unconstrained minimization problem, we wish to find an iterate  $x_k$  such that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

In the analysis of constrained problems, we may define  $\pi(k, x_k)$  to mean that  $x_k$  asymptotically approaches a constrained first-order critical point, a more general property. We will even consider<sup>111</sup> a case where only a subset of the first-order criticality conditions are required to hold asymptotically. Thus the variety of properties that can be represented by  $\pi$  is quite large, and it is very remarkable that the limit (8.1.1) may be obtained for very general criticality measures, as we will see in this section. We define  $\pi(k, x_k)$  to be a general *first-order criticality measure* of the iterate  $x_k$  if it is a nonnegative real function of its second argument such that

$$\|x_k - x_p\| \rightarrow 0 \text{ implies that } |\pi(k, x_k) - \pi(p, x_p)| \rightarrow 0 \quad (8.1.2)$$

and if the limit (8.1.1) corresponds to asymptotically satisfying the first-order criticality conditions of some optimization problem. Note that (8.1.2) can be interpreted as a continuity property of the measure across different iterates.

If we temporarily return to the unconstrained minimization problem,  $\pi(k, x_k) = \|\nabla_x f(x_k)\|$  is obviously one of many suitable criticality measures.<sup>112</sup> For this case, these may be viewed as forcing functions of  $\|\nabla_x f(x)\|$  (see Section 3.3.3). In particular, such measures include variations on the definition of the norm (in  $\|\nabla_x f(x)\|$ ): for instance, we could consider the choice

$$\pi(k, x_k) = \|\nabla_x f(x_k)\|^\psi$$

for some  $\psi > 0$  or

$$\pi(k, x_k) = \|\nabla_x f(x_k)\|_k$$

whenever the norms  $\|\cdot\|_k$  are uniformly equivalent (that is, AN.1 holds), or even

$$\pi(k, x_k) = \|\nabla_x f(x_k)\| \min \left[ \epsilon_1, \epsilon_2 \|\nabla_x f(x_k)\| \right], \quad (8.1.3)$$

where  $\epsilon_1$  and  $\epsilon_2$  are positive constants. We will see later<sup>113</sup> that the latter form is in fact very useful. It is illustrated in Figure 8.1.1 for  $\epsilon_1 = 0.0001$  and  $\epsilon_2 = 0.1$ . We will also meet other formulations of criticality measures in Sections 8.3, 12.1.3, and 13.12.

Finally, observe also that, since  $\|g_k\|$  does not play any explicit role in Algorithm BTR (p. 116), no modification of the algorithm itself is necessary if we wish to introduce a more general first-order measure  $\pi(k, x_k)$ . Ideally perhaps, the convergence analysis of Chapter 6 might have been explicitly developed around the notion of a general first-order measure instead of the simpler, but more particular,  $\|g_k\|$ . We have consciously not followed this course from the outset as it would have most likely reinforced the abstract nature of the analysis at the expense of direct intuition. However, we now devote the remains of this section to a discussion of the modifications that are required to cover the more general case.

<sup>111</sup>In Section 13.12.2.

<sup>112</sup>Observe that, in this case, the dependence of the measure on  $k$  is only via  $x_k$ .

<sup>113</sup>See Sections 8.1.5, 10.3.2, and 13.12.

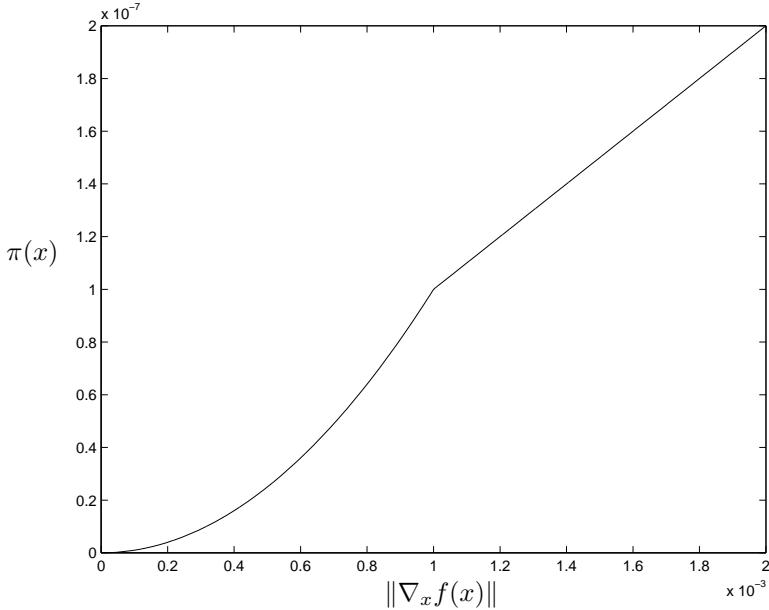


Figure 8.1.1: The criticality measure (8.1.3).

### 8.1.2 First-Order Convergence Theory

Of course, the key to the use of  $\pi(k, x_k)$  in the convergence theory is that decreasing the model can be interpreted as reasonable progress towards satisfying (8.1.1). Thus we have to rephrase AA.1, whose role, we may recall, is to guarantee that the progress that can be made at iteration  $k$  on the model  $m_k$  is at least proportional to the amount by which  $x_k$  is not first-order critical, or, in our new framework, proportional to the amount by which  $x_k$  causes (8.1.1) to be violated. If we now replace  $\|g_k\|$  by the first-order measure

$$\pi_k \stackrel{\text{def}}{=} \pi(k, x_k)$$

and define

$$\beta_k^\pi \stackrel{\text{def}}{=} 1 + \max_{x \in \mathcal{B}_k} \max_{w \neq 0} \frac{|\langle w, \nabla_{xx} m_k(x) w \rangle|}{\|w\|_k^2}, \quad (8.1.4)$$

we may rewrite AA.1 as follows.

**AA.1b** For all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \pi_k \min \left[ \frac{\pi_k}{\beta_k^\pi}, \Delta_k \right]$$

for some constant  $\kappa_{\text{mdc}} \in (0, 1)$ .

Note that we have replaced  $\beta_k$  (as defined in (6.3.1) [p. 124]) by (8.1.4). This is not the most general choice possible, as the convergence theory does not require much of this quantity. In fact, any bounded<sup>114</sup> sequence of numbers, possibly depending on

<sup>114</sup>Or, at least, slowly diverging to infinity (see Section 8.4).

the measure  $\pi(k, x)$ , would be adequate for the purpose of proving convergence results. However, this level of generality will not be used in this book. We thus consider (8.1.4) because it arises naturally in the context of dual norms, which are discussed in the next section. It allows for a simple but useful generalization of AM.4.

**AM.4b** There exists a constant  $\kappa_{\text{umh}} \geq 1$  such that

$$|\langle w, \nabla_{xx} m_k(x)w \rangle| \leq (\kappa_{\text{umh}} - 1)\|w\|_k^2$$

for all  $k$ , all  $x \in \mathcal{B}_k$ , and all  $w \neq 0$ .

Observe that AM.4b is equivalent to AM.4 when all considered norms are uniformly equivalent (i.e., when AN.1 holds). But this latter assumption is not necessary for our development, and the added flexibility obtained by abandoning it will later turn out to be useful.<sup>115</sup> In fact, we only need a “one-sided” version of the norm equivalence property for what we have in mind, which we state as follows.

**AN.1b** There exists  $\kappa_{\text{une}} > 1$  such that

$$\|w\| \leq \kappa_{\text{une}}\|w\|_k$$

for all  $w \in \mathbb{R}^n$  and all  $k$ .

In comparison with AN.1, AN.1b still ensures that  $\nu_k^S$  is bounded above, but no longer that  $\nu_k^C$  and  $\nu_k^E$  are bounded away from zero. It is thus considerably weaker. We also note that AM.4 implies AM.4b, if AN.1b replaces AN.1, because one still has, using the Cauchy–Schwarz inequality and AN.1b, that, for  $w \neq 0$ ,

$$\frac{|\langle w, \nabla_{xx} m_k(x)w \rangle|}{\|w\|_k^2} \leq \kappa_{\text{une}}^2 \frac{|\langle w, \nabla_{xx} m_k(x)w \rangle|}{\|w\|^2} \leq \kappa_{\text{une}}^2 \|\nabla_{xx} m_k(x)\|,$$

but the one-sided nature of this last assumption prevents us from deducing the reverse implication. Indeed, AM.4b may be rephrased as

$$\sup_{w \neq 0} \frac{|\langle w, \nabla_{xx} m_k(x)w \rangle|}{\|w\|^2} \left[ \frac{\|w\|}{\|w\|_k} \right]^2 \leq \kappa_{\text{umh}} - 1. \quad (8.1.5)$$

The key observation is that this formulation involves not only the usual curvature of the model in the direction  $w$ , expressed in the Euclidean norm (the first fraction), but also the square of

$$\nu_k^w \stackrel{\text{def}}{=} \frac{\|w\|}{\|w\|_k}$$

(the second fraction). It is thus equivalent to AM.4 if  $\nu_k^w$  is bounded away from zero for all  $w$ , which shows that AM.4 implies AM.4b if AN.1b holds, as we have already noticed, but that the converse is in general not true (unless all norms are uniformly equivalent, i.e., AN.1 holds). This one-sided implication raises the question of knowing whether the

---

<sup>115</sup>See Section 13.12, for instance.

missing equivalence could nevertheless be achieved along specific directions, or possibly a subspace, despite the fact that it may not hold for the complete space. Returning to (8.1.5), we see that, if we restrict our attention to a given direction  $w$ , the quotient might be bounded because the curvature of the model is asymptotically bounded in this direction, which is the direct generalization of AM.4. But equally, the quotient might be bounded because the growth of the curvature to infinity is balanced by the fact that  $\nu_k^w$  tends to zero, a property that only depends on the sequence  $\{\|\cdot\|_k\}$ , and not at all on the model. We therefore conclude that AM.4b merely requires that the model has asymptotically bounded curvature for the nonzero directions in the cone

$$\mathcal{K}_\epsilon = \text{cl} \left\{ w \in \mathbb{R}^n \mid \liminf_{k \rightarrow \infty} \frac{\|w\|}{\|w\|_k} \geq \epsilon \right\} \quad (8.1.6)$$

for some  $\epsilon > 0$ , which is the set of directions where  $\|\cdot\|$  and  $\|\cdot\|_k$  are uniformly equivalent with constant  $\max[\kappa_{\text{une}}, \epsilon^{-1}]$ , given AN.1b. Observe that, if AN.1 holds, then  $\mathcal{K}_\epsilon = \mathbb{R}^n$  for all  $\epsilon \leq \kappa_{\text{une}}^{-1}$ , but, in general, this cone may have a complicated geometry. For example, if  $\|w\|_k = k\|w\|$ , then  $\mathcal{K}_\epsilon = \emptyset$  for all  $\epsilon > 0$ . If

$$\|w\|_k = \begin{cases} \min \left[ k, \frac{1}{|\langle z, w \rangle|} \right] \|w\| & \text{if } \langle z, w \rangle \neq 0, \\ k\|w\| & \text{if } \langle z, w \rangle = 0 \end{cases}$$

for some  $z \neq 0$ , then  $\mathcal{K}_\epsilon$  is a revolution cone around  $z$  whose “opening angle” depends on  $\epsilon$ . If

$$\|w\|_k^2 = \left\langle w, \left[ e_j e_j^T + k \sum_{i \neq j} e_i e_i^T \right] w \right\rangle, \quad \text{with } j = k \bmod n,$$

then  $\mathcal{K}_\epsilon$  consists of all multiples of the vectors of the canonical basis, for all  $\epsilon > 0$ . But the geometry of  $\mathcal{K}_\epsilon$  can also be milder, and, for our present purposes, more interesting. For instance, if

$$\|w\|_k^2 = \langle w, (I + kA^T A)w \rangle,$$

then  $\mathcal{K}_\epsilon$  is the null-space of the matrix  $A$  for all  $\epsilon > 0$ , a property that we will later exploit in the context of constrained problems.

Having reviewed our assumptions, we are now ready to reexamine the theory of Chapter 6, taking into account the substitutions

$$\begin{array}{ccc} \|g_k\| & \longrightarrow & \pi_k, \\ \text{AM.4, AA.1} & \longrightarrow & \text{AM.4b, AA.1b}, \\ \text{AN.1} & \longrightarrow & \text{AN.1b}, \\ x_* \text{ first-order critical} & \longrightarrow & \lim_{k \rightarrow \infty} \pi_k = 0. \end{array} \quad (8.1.7)$$

**Theorem 6.4.1:** This result continues to hold without any modification. The only modification to the proof arises in (6.4.7), where we have now, using AM.4b

instead of AM.4, that

$$\begin{aligned}
 |f(x_k + s_k) - m_k(x_k + s_k)| &= \frac{1}{2} |\langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle - \langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\
 &\leq \frac{1}{2} |\langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle| + \frac{1}{2} |\langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\
 &\leq \frac{1}{2} \kappa_{\text{ufh}} \|s_k\|^2 + (\kappa_{\text{umh}} - 1) \|s_k\|_k^2 \\
 &= \frac{1}{2} (\kappa_{\text{ufh}} [\nu_k^S]^2 + \kappa_{\text{umh}} - 1) \|s_k\|_k^2 \\
 &= \frac{1}{2} (\kappa_{\text{ufh}} [\nu_k^S]^2 + \kappa_{\text{umh}} - 1) \Delta_k^2.
 \end{aligned}$$

We may then use AN.1b (instead of AN.1) and deduce (6.4.3).

**Theorem 6.4.2:** This is the first time where we have to replace  $\|g_k\|$  by  $\pi_k$ . To verify that the theorem remains true in our context, we merely make this substitution and use AA.1b instead of AA.1 in order to obtain (6.4.11).

**Theorem 6.4.3:** No change is necessary here, except that the initial assumption that  $\|g_k\| \geq \kappa_{\text{lbg}}$  must now be replaced by  $\pi_k \geq \kappa_{\text{lbg}}$ .

**Theorem 6.4.4:** As in the original proof, we deduce from the assumptions that the iterates are all identical for large  $k$  and that the trust-region radius converges to zero. If  $\pi_k$  remains bounded away from zero, we still deduce a contradiction with Theorem 6.4.2 and therefore obtain that  $\pi_k = 0$  for all  $k$  sufficiently large, which corresponds to the desired conclusion.

**Theorem 6.4.5:** As above, we still obtain the conclusion that

$$\liminf_{k \rightarrow \infty} \pi_k = 0 \quad (8.1.8)$$

simply by substituting  $\pi_k$  for  $\|g_k\|$  and using AA.1b instead of AA.1 and AM.4b instead of (6.3.2) when deriving (6.4.17).

**Theorem 6.4.6:** Again, we simply substitute  $\pi_k$  for  $\|g_k\|$ , AA.1b for AA.1, and AM.4b instead of (6.3.2), and deduce (6.4.21). In order to derive (6.4.22), we have to call upon AN.1b (instead of AN.1) to show that the  $\nu_j^S$  are bounded above. We may then conclude the proof without difficulty, taking into account the crucial observation that the continuity of the gradient must now be replaced by (8.1.2), which ensures that

$$\lim_{i \rightarrow \infty} |\pi_{t_i} - \pi_{\ell_i}| = 0 \quad \text{whenever} \quad \lim_{i \rightarrow \infty} \|x_{t_i} - x_{\ell_i}\| = 0,$$

as necessary to obtain the desired contradiction. We thus deduce that

$$\lim_{k \rightarrow \infty} \pi_k = 0. \quad (8.1.9)$$

This last result is the equivalent of  $\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0$ , and we therefore see that we have extended the complete convergence theory for first-order critical points to our considerably more general framework. It is important to note that condition (8.1.2) in the definition of the first-order measure has not been used before considering Theorem 6.4.6, which means that we may prove (8.1.8) even without this assumption.

### 8.1.3 Second-Order Convergence Theory

We now investigate if the results of Chapter 6 concerning convergence to second-order critical points may also be extended to our general framework. Unfortunately the proof of Theorem 6.5.2 does not seem to extend, because we have not assumed differentiability of our first-order criticality measure, and, even if this was the case, its link with convexity would be unclear. However, if one is ready to incorporate in the algorithm, as we did in AA.2, the requirement that some model reduction can be obtained when the model is not convex, then it is possible to obtain results that generalize those of Section 6.6. Because this generalization will be useful later, we discuss it in some detail.

We first need to adapt our assumptions on the Hessian of the model and exploitation of negative curvature. As we have used  $\beta_k^\pi$  in AA.1b, which represents the curvature of the model as seen in the norm  $\|\cdot\|_k$ , we continue to use the same approach.

**AM.5b** We assume that

$$\lim_{k \rightarrow \infty} \sup_{w \neq 0} \frac{|\langle w, [\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)]w \rangle|}{\|w\|_k^2} = 0$$

whenever  $\lim_{k \rightarrow \infty} \pi_k = 0$ .

If we translate our reasoning concerning AM.4b to this case, we see that AM.5b only imposes a requirement that the Hessians of the model and objective function asymptotically coincide in  $\mathcal{K}_\epsilon$  for each  $\epsilon > 0$  for which this set is nonempty. Thus, as above, AM.5 implies AM.5b, but the converse is in general not true unless one assumes AN.1. We also note that, as was the case for AM.5, AM.5b and AF.1 together imply AM.4b if the iterates remain in a bounded domain.

We also rephrase our assumption that the Hessian of the model is Lipschitz continuous.

**AM.6b** There exists a constant  $\kappa_{\text{lch}} > 0$  such that

$$\sup_{w \neq 0} \frac{|\langle w, [\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)]w \rangle|}{\|w\|_k^2} \leq \kappa_{\text{lch}} \|x - y\|_k$$

for all  $x, y \in \mathcal{B}_k$ .

Again, AM.6 implies AM.6b because of AN.1b, but the converse may not be true. From our discussion above, we see that AM.5b amounts to assuming Lipschitz continuity of the Hessian of the model in  $\mathcal{K}_\epsilon$ , if this set is not empty for some  $\epsilon > 0$ .

**AA.2b** If

$$\tau_k \stackrel{\text{def}}{=} \inf_{w \neq 0} \frac{\langle w, \nabla_{xx} m_k(x_k)w \rangle}{\|w\|_k^2} < 0,$$

then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min[\tau_k^2, \Delta_k^2, \kappa_{\text{lsd}}^2]$$

for some constants  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$  and<sup>116</sup>  $\kappa_{\text{lsd}} > 0$ .

---

<sup>116</sup>“lsd” stands for “limit of small deltas”.

This is very similar to AA.2, except that the constant  $\kappa_{\text{lsd}}$  has been introduced as an extra quadratic term multiplying the absolute value of the negative eigenvalue. We would not expect this modification to substantially alter the results obtained, since AA.2b is typically used in a context where  $\Delta_k$  converges to zero, in which case the  $\kappa_{\text{lsd}}$  term will asymptotically be irrelevant. This modification will turn out to be useful in Section 13.3, where we study barrier methods for problems with convex inequality constraints. It is also useful to understand the meaning of  $\tau_k$ , given our observations of the norms  $\|\cdot\|_k$ , particularly since this quantity plays an important role in the results that follow.

**Lemma 8.1.1** Suppose that AM.1 and AN.1b hold and let  $\{x_{k_i}\}$  be a subsequence of the iterates. Then the limit

$$\liminf_{i \rightarrow \infty} \tau_{k_i} \geq 0 \quad (8.1.10)$$

implies that, for all  $\epsilon > 0$ ,

$$\liminf_{i \rightarrow \infty} \inf_{\substack{w \in \mathcal{K}_\epsilon \\ w \neq 0}} \frac{\langle w, \nabla_{xx} m_{k_i}(x_{k_i}) w \rangle}{\|w\|^2} \geq 0$$

when the set  $\mathcal{K}_\epsilon$  is nonempty. Moreover, if, for some  $\epsilon > 0$ ,  $\mathcal{K}_\epsilon$  is a subspace of which the columns of the matrix  $Z_\epsilon$  form a basis, the limit (8.1.10) then implies that

$$\liminf_{i \rightarrow \infty} \lambda_{\min}[Z_\epsilon^T \nabla_{xx} m_{k_i}(x_{k_i}) Z_\epsilon] \geq 0.$$

**Proof.** The first part results from our discussion after AN.1b. The second simply results from applying the first in the more specific case of a subspace and using the relation between eigenvalues and Rayleigh quotients in that subspace.  $\square$

Returning to AA.2b, we may thus interpret it as assuming that negative curvature in  $\mathcal{K}_\epsilon$  is exploited when present for any  $\epsilon > 0$ .

If we now consider extending the results of Section 6.6, we then see that this extension involves, besides (8.1.7), the substitutions

$$\begin{array}{lll} \nabla_{xx} f(x) & \longrightarrow & \forall \epsilon > 0, \quad Z_\epsilon^T \nabla_{xx} f(x) Z_\epsilon, \\ \text{AM.5, AM.6, AA.2} & \longrightarrow & \text{AM.5b, AM.6b, AA.2b} \\ x_* \text{ second-order critical} & \longrightarrow & \lim_{k \rightarrow \infty} \pi_k = 0 \quad \forall \epsilon > 0 \quad \lambda_{\min}[Z_\epsilon^T \nabla_{xx} f(x_*) Z_\epsilon] \geq 0, \end{array}$$

where we have assumed, for simplicity, that for all  $\epsilon > 0$ ,  $\mathcal{K}_\epsilon$  is a subspace of  $\mathbb{R}^n$  spanned by the columns of  $Z_\epsilon$ .

In Lemma 6.5.3 (p. 144), as for Theorem 6.4.1 (p. 133), we have to use the quotients of AM.5b instead of the norms of AM.5, which implies that the constant  $\kappa_{\text{une}}^2$  no longer appears in the bound on  $|\rho_{k_i} - 1|$ . Lemma 6.5.4 (p. 145) and Theorem 6.6.3 (p. 154) are

unmodified. For future reference, we decompose the proof of Theorem 6.6.4 (p. 154) in two successive steps. We first show that at least a subsequence of the  $\tau_k$  converges to a positive limit.

**Theorem 8.1.2** Suppose that AF.1–AF.3, AN.1b, AM.1–AM.3, AM.4b–AM.6b, AA.1b, and AA.2b hold. Then

$$\limsup_{k \rightarrow \infty} \tau_k \geq 0,$$

where  $\tau_k$  is defined in AA.2b.

**Proof.** The proof proceeds exactly as for Theorem 6.6.4, where we now use the modified assumptions, except that we have to start from the assumption that there exists  $\lambda_* > 0$  such that  $\tau_k \leq -\frac{1}{2}\lambda_*$  for  $k$  sufficiently large in order to derive a contradiction. The introduction of  $\kappa_{\text{lsd}}$  then implies that (6.6.21) (p. 154) becomes

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}\kappa_{\text{sod}}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \Delta_k^2, \kappa_{\text{lsd}}^2].$$

Then we select  $\delta_1 \leq \min[\frac{1}{2}|\lambda_*|, \kappa_{\text{lsd}}]$ , (6.6.23) (p. 155) becomes

$$f(x_{k_0+j}) - f(x_{k_0+j+1}) \geq \frac{1}{2}\eta_1\kappa_{\text{sod}}|\lambda_*| \min[\frac{1}{4}\lambda_*^2, \delta_2^2, \kappa_{\text{lsd}}^2] > 0,$$

and the rest of the proof is unmodified.  $\square$

Combining this result and Lemma 8.1.1, we then deduce that the models are asymptotically convex (along a subsequence) *on*  $\mathcal{K}_\epsilon$ . As a consequence, we obtain the following rewording of Theorem 6.6.4.

**Theorem 8.1.3** Suppose that AF.1–AF.3, AN.1b, AM.1–AM.3, AM.4b–AM.6b, AA.1b, and AA.2b hold. Then

$$\limsup_{k \rightarrow \infty} \inf_{\substack{w \in \mathcal{K}_\epsilon \\ w \neq 0}} \frac{\langle w, \nabla_{xx} f(x_k) w \rangle}{\|w\|^2} \geq 0$$

for all  $\epsilon > 0$ . If, additionally, for some  $\epsilon > 0$ ,  $\mathcal{K}_\epsilon$  is a subspace of which the columns of the matrix  $Z_\epsilon$  form a basis, then

$$\limsup_{k \rightarrow \infty} \lambda_{\min}[Z_\epsilon^T \nabla_{xx} f(x_k) Z_\epsilon] \geq 0.$$

**Proof.** The result directly follows from Theorem 8.1.2, Lemma 8.1.1, and AM.5b.  $\square$

Theorem 6.6.5 extends without difficulty to ensure that  $\lim_{k \rightarrow \infty} \pi_k = 0$  and the objective function is convex at  $x_*$  for all nonzero directions  $w \in \mathcal{K}_\epsilon$  for all  $\epsilon > 0$ . The proof of Lemma 6.6.6 (p. 156) also extends in a straightforward way, so long as one notices that (6.6.25) simply becomes  $\tau_{k_i} < \frac{1}{2}\lambda_*$  and remembers that the term  $\kappa_{\text{lsd}}^2$  must be introduced into the minima of (6.6.28) (p. 156). The generalization of Theorem 6.6.7 is again very easy if one starts the proof by assuming that there is an  $\epsilon > 0$  such that the objective function has negative curvature in  $\mathcal{K}_\epsilon$  at  $x_*$ . In the course of the proof, we also have to replace (6.6.34) (p. 158) by

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min[\tau_k^2, \Delta_k^2, \kappa_{\text{lsd}}^2] \geq \frac{1}{2} \kappa_{\text{sod}} \kappa_{\text{snc}} |\lambda_*| \Delta_k^2 > 0,$$

where we also use the fact that  $\Delta_k$  converges to zero (see (6.6.32)) to obtain the second inequality. We thus obtain the following statement.

**Theorem 8.1.4** Suppose that AF.1–AF.3, AN.1b, AM.1–AM.3, AM.4b–AM.6b, AA.1b, and AA.2b hold, and also that  $x_*$  is an isolated limit point of the sequence of iterates  $\{x_k\}$ . Then  $\lim_{k \rightarrow \infty} \pi_k = 0$  and  $\lambda_{\min}[Z_\epsilon^T \nabla_{xx} f(x_*) Z_\epsilon] \geq 0$  for all  $\epsilon > 0$ .

If we also assume AA.3, the final convergence result may be generalized in the following form.

**Theorem 8.1.5** Suppose that AF.1–AF.3, AN.1b, AM.1–AM.3, AM.4b–AM.6b, AA.1b, AA.2b, and AA.3 hold. Then, for every convergent subsequence  $\{x_{k_i}\}$ ,

$$\liminf_{i \rightarrow \infty} \tau_{k_i} \geq 0$$

and

$$\liminf_{i \rightarrow \infty} \lambda_{\min}[Z_{k_i}^T \nabla_{xx} f(x_{k_i}) Z_{k_i}] \geq 0$$

for all  $\epsilon > 0$ .

**Proof.** The proof of the first claim is very similar to that of Theorem 6.6.8 (p. 159), with the proper substitution of assumptions. We now assume  $\tau_{k_i} \leq -\frac{1}{2}\lambda_* < 0$  for  $k$  sufficiently large, from which we deduce (6.6.37) (p. 159) instead of (6.6.36) for obtaining the contradiction. We then deduce from AA.2b that

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \Delta_k^2, \kappa_{\text{lsd}}^2]$$

instead of (6.6.38). We then consider the ratio  $\rho_k$  for  $\Delta_k \leq \min[\frac{1}{2}|\lambda_*|, \kappa_{\text{lsd}}]$  and deduce that

$$f(x_{k+j}) - f(x_{k+j+1}) \geq \frac{1}{2} \eta_1 \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_2^2, \kappa_{\text{lsd}}^2] > 0$$

instead of (6.6.41) (p. 160). The rest of the proof follows if we replace (6.6.44) by

$$f(x_p) - f(x_{q+1}) \geq f(x_j) - f(x_{j+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min \left[ \frac{1}{4} \lambda_*^2, \delta_4^2, \kappa_{\text{lsd}}^2 \right] \stackrel{\text{def}}{=} \delta_3 > 0,$$

(6.6.46) (p. 160) by

$$f(x_p) - f(x_{q+1}) \geq f(x_{j-1}) - f(x_j) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min \left[ \frac{1}{4} \lambda_*^2, \left( \frac{\delta_4}{\gamma_4} \right)^2, \kappa_{\text{lsd}}^2 \right] \stackrel{\text{def}}{=} \delta_5 > 0,$$

and (6.6.48) (p. 161) by

$$f(x_p) - f(x_{q+1}) \geq f(x_q) - f(x_{q+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min \left[ \frac{1}{4} \lambda_*^2, \delta_6^2, \kappa_{\text{lsd}}^2 \right] \stackrel{\text{def}}{=} \delta_7 > 0.$$

The second claim follows from our discussion above and the definition of  $\mathcal{K}_\epsilon$ .  $\square$

We recall here that our restriction to the case where  $\mathcal{K}_\epsilon$  is a subspace is purely for notational convenience and that all the above results trivially extend to the case where the geometry of this cone is more complex.

If we now consider extending the results of Section 6.5, then, unfortunately, AN.1b no longer suffices.<sup>117</sup> The main difficulty is that AN.1b and the assumptions on  $\pi_k$  do not easily allow us to relate the norm of steps or gradients at successive iterates, and this prevents us from proving the equivalent of Theorem 6.5.2 (p. 141). Furthermore, the link between  $\pi_k$  and the length of the step along a positive curvature direction (as stated by Lemma 6.5.1 [p. 140]) also vanishes. Since this result is crucial in Theorem 6.5.5 (p. 146), the generalization of this last result also eludes us. On the other hand, if we return to the simpler case where  $\pi_k$  is a first-order criticality measure and AN.1 holds, then these difficulties disappear and all the results of Section 6.5 may be extended.

### 8.1.4 Convergence in Dual Norms

As a first example of an application of the general framework we have just developed, we consider convergence properties expressed in a new norm. In fact, the redefinition of the norm in the space of the problem's variables, which we discussed in Sections 6.1 and 6.7, is not without further consequences. One of them is that, intuitively, one might expect that measuring criticality in the scaled norm  $\|S_k^T g_k\|$  (see (6.7.3) [p. 164]) would be more appropriate, as it better balances the contributions of the variables in the definition of the local slope. In other words, we would like to consider the sequence of iterates in the scaled space, instead of considering them in the original unscaled space. We devote this section to an investigation of the convergence properties of Algorithm BTR (p. 116) in this context. More generally, if the trust region is defined, at iteration  $k$ , by the norm  $\|\cdot\|_k$ , we now consider measuring the gradient  $g_k$  using  $\|g_k\|_{[k]}$ , where  $\|\cdot\|_{[k]}$  is the dual norm of  $\|\cdot\|_k$ , which is defined by

$$\|y\|_{[k]} \stackrel{\text{def}}{=} \sup_{x \neq 0} \frac{|\langle y, x \rangle|}{\|x\|_k} \quad (8.1.11)$$

---

<sup>117</sup>At least as far as we know.

(see Section 2.3.1). Note that this ensures that the Hölder inequality

$$|\langle y, x \rangle| \leq \|x\|_k \|y\|_{[k]} \quad (8.1.12)$$

holds for all  $x, y \in \mathbb{R}^n$ . Associated with the dual norm, we also define the corresponding operator norm, the *bidual* norm

$$\|H\|_{\{k\}} \stackrel{\text{def}}{=} \sup_{y \neq 0} \frac{\|Hy\|_{[k]}}{\|y\|_k}, \quad (8.1.13)$$

for which we easily derive that, for all  $x, y \in \mathbb{R}^n$ ,

$$|\langle y, Hx \rangle| = |\langle Hy, x \rangle| \leq \|Hy\|_{[k]} \|x\|_k \leq \|H\|_{\{k\}} \|y\|_k \|x\|_k. \quad (8.1.14)$$

We briefly summarize the properties of the dual norm, in the case where  $\|\cdot\|_k$  is the (scaled) ellipsoidal norm (6.7.9) (p. 165).

**Lemma 8.1.6** Suppose that

$$\|x\|_k = \sqrt{\langle x, S_k^{-T} S_k^{-1} x \rangle} = \|S_k^{-1} x\| = \|x\|_{S_k^{-T} S_k^{-1}}$$

for some nonsingular scaling matrix  $S_k$ . Then its dual norm is given, for any  $y$ , by

$$\|y\|_{[k]} = \sqrt{\langle y, S_k S_k^T y \rangle} = \|S_k^T y\| = \|y\|_{S_k S_k^T} \quad (8.1.15)$$

and

$$\|H\|_{\{k\}} = \|S_k^T H S_k\|. \quad (8.1.16)$$

**Proof.** From the definition (8.1.11) and the Cauchy–Schwarz inequality, we have that

$$\|y\|_{[k]} = \sup_{x \neq 0} \frac{|\langle y, x \rangle|}{\|S_k^{-1} x\|} = \sup_{x \neq 0} \frac{|\langle S_k^T y, S_k^{-1} x \rangle|}{\|S_k^{-1} x\|} = \sup_{z \neq 0} \frac{|\langle S_k^T y, z \rangle|}{\|z\|} = \|S_k^T y\|$$

and (8.1.15) follows. Furthermore, we deduce from the definitions (8.1.13) and (8.1.11) and the properties of the  $\ell_2$  norm that

$$\begin{aligned} \|H\|_{\{k\}} &= \sup_{x \neq 0} \frac{\|Hx\|_{[k]}}{\|x\|_k} \\ &= \sup_{x \neq 0} \sup_{y \neq 0} \frac{|\langle y, Hx \rangle|}{\|y\|_k \|x\|_k} \\ &= \sup_{x \neq 0} \sup_{y \neq 0} \frac{|\langle S_k^{-1} y, [S_k^T H S_k] S_k^{-1} x \rangle|}{\|S_k^{-1} y\| \|S_k^{-1} x\|} \\ &= \sup_{u \neq 0} \sup_{v \neq 0} \frac{|\langle u, [S_k^T H S_k] v \rangle|}{\|u\| \|v\|} \\ &= \|S_k^T H S_k\|, \end{aligned}$$

which proves (8.1.16).  $\square$

The use of the dual norm (8.1.11) thus extends our means of adequately measuring the size of the gradient beyond a simple rescaling of the variables to a more general redefinition of the geometric shape of the trust region.

We next analyse a little further the consequences of AN.1b in the framework of dual norms.

**Theorem 8.1.7** Suppose that AN.1b holds. Then, for all  $x \in \mathbb{R}^n$ ,

$$\|x\|_{[k]} \leq \kappa_{\text{une}} \|x\|. \quad (8.1.17)$$

Furthermore, we have that

$$\|H\|_{\{k\}} \leq \kappa_{\text{une}}^2 \|H\| \quad (8.1.18)$$

for every symmetric  $n \times n$  matrix  $H$ .

**Proof.** The bound (8.1.17) immediately follows from the inequality

$$\|x\|_{[k]} = \max_{y \neq 0} \frac{|\langle y, x \rangle|}{\|y\|_k} \leq \frac{\|y\|}{\|y\|_k} \|x\| \leq \kappa_{\text{une}} \|x\|,$$

where we used the definition of the dual norm, the Cauchy–Schwarz inequality, and AN.1b. Finally, (8.1.18) follows from

$$\|H\|_{\{k\}} \stackrel{\text{def}}{=} \sup_{y \neq 0} \frac{\|Hy\|_{[k]}}{\|y\|_k} \leq \sup_{y \neq 0} \frac{\kappa_{\text{une}} \|Hy\|}{\|y\|_k} \leq \sup_{y \neq 0} \frac{\kappa_{\text{une}} \|H\| \|y\|}{\|y\|_k} \leq \kappa_{\text{une}}^2 \|H\|,$$

where we used (8.1.17) and AN.1b.  $\square$

We immediately notice that (8.1.17) makes it possible to have that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\|_{[k]} = 0,$$

while  $\|\nabla_x f(x_k)\|$  remains bounded away from zero for all  $k$ . Hence convergence in the dual norm must not be automatically interpreted as convergence to an unconstrained first-order critical point, which is perhaps a surprising result. This observation indicates that, if we wish to apply the extended convergence framework discussed earlier in this section, we cannot simply consider  $\|g_k\|_{[k]}$  as a criticality measure, but have to consider it as a first-order measure (at least if AN.1 does not hold). What are the other properties that we should verify if we wish to apply this framework? The first is obviously to show that  $\|g_k\|_{[k]}$  satisfies the conditions we have imposed on first-order measures and, in particular, that it satisfies (8.1.2). Unfortunately, this may not be true in general, because AN.1b leaves too much freedom for  $\|\cdot\|_k$  to differ, possibly wildly, from  $\|\cdot\|_t$ . Thus we have to make a further assumption.

**AN.3** If  $x_k$  and  $x_t$  are two iterates of the sequence  $\{x_k\}$ , we have that

$$\left| \|g_k\|_{[k]} - \|g_t\|_{[t]} \right| \rightarrow 0 \text{ whenever } \|x_k - x_t\| \rightarrow 0.$$

This assumption is satisfied, for instance, when the scaling is constant, that is, when  $\|\cdot\|_k$  does not depend on  $k$ , or when AN.1 holds, as is shown by the following easy proposition.

**Theorem 8.1.8** Suppose that AN.1 holds and that  $\|\cdot\|_{[k]}$  is defined according to (8.1.11). Then AN.1 also holds for the dual norms  $\|\cdot\|_{[k]}$ . Furthermore, if AF.1 and AM.3 are also satisfied, then AN.3 holds.

**Proof.** We have that, for any  $y \in \mathbb{R}^n$  and any  $k$ ,

$$\|y\|_{[k]} = \max_{\|x\|_k=1} |\langle x, y \rangle| \leq \frac{\|x\|}{\|x\|_k} \|y\| \leq \kappa_{\text{une}} \|y\|, \quad (8.1.19)$$

where we have used the continuity of the norm and the compactness of the unit ball in  $\mathbb{R}^n$  to derive the equality, the Cauchy–Schwarz inequality to obtain the first inequality, and AN.1 to obtain the second. On the other hand, the maximum in this latter relation is larger than its value for  $x = y/\|y\|_k$ , yielding, because of AN.1,

$$\|y\|_{[k]} = \max_{\|x\|_k=1} |\langle x, y \rangle| \geq \frac{\|y\|}{\|y\|_k} \|y\| \geq \frac{1}{\kappa_{\text{une}}} \|y\|. \quad (8.1.20)$$

We thus obtain from (8.1.19) and (8.1.20) that AN.1 holds for the dual norms (with the same equivalence constant). The fact that AN.3 holds may then be easily deduced from the fact that the dual norms  $\|\cdot\|_{[k]}$  are now uniformly equivalent to the Euclidean norm and that the gradient of the model is continuous because of AF.1 and AM.3.  $\square$

We will see later<sup>118</sup> that AN.3 is also satisfied in at least one other case of interest.

Now that we have a proper first-order measure ( $\|g_k\|_{[k]}$ ) and proper assumptions on its continuity across iterations and the norm (AN.3 and AN.1b), our main concern is to analyse if a condition on sufficient model reduction can be obtained for this measure. We show that this is the case by reexamining the model decrease at the Cauchy point.

**Theorem 8.1.9** If the model is quadratic and given by (6.1.7) (p. 117), and if we define the Cauchy point by exact minimization along the Cauchy arc, that is, (6.3.3) (p. 124), we have that

$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2} \|g_k\|_{[k]} \min \left[ \frac{\|g_k\|_{[k]}}{\beta_{\{k\}}}, \Delta_k \right], \quad (8.1.21)$$

where

$$\beta_{\{k\}} \stackrel{\text{def}}{=} 1 + \max_{x \in \mathcal{B}_k} \sup_{w \neq 0} \frac{|\langle w, \nabla_{xx} m_k(x) w \rangle|}{\|w\|_k^2}. \quad (8.1.22)$$

---

<sup>118</sup>In Section 13.12.

**Proof.** The proof is essentially identical to that of Theorem 6.3.1 (p. 125), except that we now use (8.1.12) instead of the Cauchy–Schwarz inequality and, most importantly, that  $\nu_k^C$  has disappeared. We restate it in full for completeness. We first note that, for all  $t \geq 0$ ,

$$m_k(x_k - tg_k) = m_k(x_k) - t\|g_k\|_k\|g_k\|_{[k]} + \frac{1}{2}t^2\langle g_k, H_k g_k \rangle. \quad (8.1.23)$$

We now consider the case where the curvature of the model along the steepest descent direction is positive, that is, when

$$\langle g_k, H_k g_k \rangle > 0, \quad (8.1.24)$$

and compute the value of the parameter  $t$  at which the unique minimum of (8.1.23) is attained. Let us denote this optimal parameter by  $t_k^*$ . Differentiating (8.1.23) with respect to  $t$  and equating the result to zero, we obtain that

$$0 = \|g_k\|_k\|g_k\|_{[k]} - t_k^*\langle g_k, H_k g_k \rangle,$$

which immediately gives that

$$t_k^* = \frac{\|g_k\|_k\|g_k\|_{[k]}}{\langle g_k, H_k g_k \rangle}. \quad (8.1.25)$$

Two subcases may then occur. The first is when this minimizer lies within the trust region, that is, when  $t_k^*\|g_k\|_k \leq \Delta_k$ . Then  $t_k^C = t_k^*$  and we may replace this expression in the model decrease (8.1.23), which allows us to deduce that

$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2} \frac{\|g_k\|_k^2\|g_k\|_{[k]}^2}{\langle g_k, H_k g_k \rangle} \geq \frac{\|g_k\|_{[k]}^2}{2\beta_{\{k\}}}, \quad (8.1.26)$$

where we used (8.1.14) and (8.1.22). If, on the other hand,

$$t_k^*\|g_k\|_k > \Delta_k, \quad (8.1.27)$$

then the line minimum is outside the trust region and we have that

$$t_k^C\|g_k\|_k = \Delta_k. \quad (8.1.28)$$

Combining (8.1.25), (8.1.27), and (8.1.28), we see that

$$\langle g_k, H_k g_k \rangle \leq \frac{\|g_k\|_k\|g_k\|_{[k]}}{t_k^C}.$$

Substituting this last inequality in (8.1.23) and using (8.1.28), we obtain that

$$\begin{aligned} m_k(x_k) - m_k(x_k^C) &= t_k^C\|g_k\|_k\|g_k\|_{[k]} - \frac{1}{2}[t_k^C]^2\langle g_k, H_k g_k \rangle \\ &\geq \|g_k\|_{[k]}\Delta_k - \frac{1}{2}\|g_k\|_{[k]}\Delta_k \\ &= \frac{1}{2}\|g_k\|_{[k]}\Delta_k. \end{aligned} \quad (8.1.29)$$

Finally, we consider the case where the curvature of the model along the steepest-descent direction is negative, that is, when (8.1.24) is violated. We then obtain from (8.1.23) that

$$\begin{aligned} m_k(x_k - tg_k) &= m_k(x_k) - t\|g_k\|_k\|g_k\|_{[k]} + \frac{1}{2}t^2\langle g_k, H_k g_k \rangle \\ &\leq m_k(x_k) - t\|g_k\|_k\|g_k\|_{[k]} \end{aligned} \quad (8.1.30)$$

for all  $t \geq 0$ . In that case, it is easy to see that the Cauchy point must lie on the boundary of the trust region, and thus that (8.1.28) holds. Combining this equality and (8.1.30), we deduce that

$$m_k(x_k) - m_k(x_k^C) \geq \|g_k\|_k\|g_k\|_{[k]} \frac{\Delta_k}{\|g_k\|_k} = \|g_k\|_{[k]}\Delta_k \geq \frac{1}{2}\|g_k\|_{[k]}\Delta_k. \quad (8.1.31)$$

We may then conclude our proof by noting that (8.1.26), (8.1.29), and (8.1.31) imply that (8.1.21) holds.  $\square$

A similar variant of Theorem 6.3.1 (p. 125) can also be proved for the case where the model is not quadratic, but we shall not cover this in detail. However, Theorem 8.1.9 is enough to convince us that AA.1b holds in our context with  $\beta_k^\pi = \beta_{\{k\}}$  given by (8.1.22).

We may then apply the results of Section 8.1.2 and deduce that, if AF.1–AF.3, AN.1b, AM.1–AM.3, AM.4b, and AA.1b hold, then

$$\liminf_{k \rightarrow \infty} \|\nabla_x f(x_k)\|_{[k]} = 0. \quad (8.1.32)$$

Note, as we have already discussed, that (8.1.32) *does not* imply that at least a limit point is first-order critical, if the limit points exist. To see this, consider the problem of minimizing the square of the Euclidean distance to the point  $(1, -1)$  in the two-dimensional plane, and assume that a sequence of iterates  $\{x_k\}$  is generated<sup>119</sup> that converges to  $(1, 0)$ . Assume furthermore that

$$\left\| \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right\|_{[k]} = [x_k]_1|y_1| + [x_k]_2|y_2|.$$

Then it is easy to see that (8.1.32) holds but that

$$\nabla_x f \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \neq 0,$$

a situation which is of course impossible if AN.1 holds. Another similar example will be discussed in Section 13.12. This result is therefore (considerably) weaker than that of Theorem 6.4.5 (p. 136), but this is not entirely surprising since our assumptions are also weaker. Notice that we have not used AN.3 yet. If we now assume that AN.3 also holds, then the discussion of Section 8.1.2 ensures that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\|_{[k]} = 0. \quad (8.1.33)$$

---

<sup>119</sup>We are not saying it is generated by Algorithm BTR (p. 116).

Again, we have to be cautious in interpreting this result, as it does not automatically ensure that all limit points are first-order critical, if they exist. The fact that the result holds is nevertheless remarkable, given the weak nature of AN.1b and AN.3.

The definition of the eigenpoint can also be adapted to the scaled space. As above, we only consider the case where the model is quadratic but immediately indicate that similar arguments and proofs can easily be derived for general ones. We thus consider the case where  $\tau_k < 0$ , where  $\tau_k$  is defined as in AA.2b. In the special case where the scaling is linear, that is, when  $\|w\|_k = \|S_k^{-1}w\|$ , then

$$0 > \inf_{w \neq 0} \frac{\langle w, \nabla_{xx} m_k(x_k) w \rangle}{\|S_k^{-1}w\|^2} = \inf_{y \neq 0} \frac{\langle y, [S_k^T \nabla_{xx} m_k(x_k) S_k] y \rangle}{\|y\|^2},$$

which means that the Hessian of the scaled model  $S_k^T \nabla_{xx} m_k(x_k) S_k$  also has a negative eigenvalue. Let  $u_k$  be a vector such that

$$\langle g_k, u_k \rangle \leq 0, \quad \|u_k\|_k = \Delta_k, \quad \text{and} \quad \langle u_k, \nabla_{xx} m_k(x_k) u_k \rangle \leq \kappa_{\text{snc}} \tau_k \Delta_k^2$$

for some  $\kappa_{\text{snc}} \in (0, 1]$ . In the case of a linear scaling,  $S_k^{-1}u_k$  can be viewed as an approximation of an eigenvector corresponding to the most negative eigenvalue of the Hessian of the scaled model. We then calculate the eigenpoint by minimizing the model in the direction  $u_k$  while remaining in the trust region, as before. If we replace AM.4–AM.5 and AA.1 by AM.4b–AM.5b and AA.1b, Theorem 6.6.1 (p. 149) then easily extends to our more general framework, giving that

$$m_k(x_k) - m_k(x_k^E) \geq -\frac{1}{2} \kappa_{\text{snc}} \tau_k \Delta_k^2$$

(note the disappearance of  $\nu_k^E$  from the bound, a result of its disappearance from the definition of  $u_k$ ). The proof is identical to that of Theorem 6.6.1, except that we now use the Hölder inequality (8.1.12) instead of the Cauchy–Schwarz inequality. Thus, as expected, we may reformulate AM.6 and AA.2 in the form of AM.6b and AA.2b (with  $\kappa_{\text{lsd}} = \infty$ ), and therefore, using the results of Section 8.1.3, *we deduce that Theorem 8.1.2 holds and that Theorems 6.6.5, 6.6.7, and 6.6.8, as well as Lemma 6.6.6 (pp. 155–159), remain valid when the sequence of iterates is considered as belonging to the scaled space*. Moreover, as discussed above, the results of Section 6.5 also generalize if we replace AN.1b by AN.1. Furthermore, AN.3 is no longer necessary in that case, as is shown by Theorem 8.1.8. This then means that all convergence results expressed in the  $\ell_2$  norm may be recast in any of the  $\|\cdot\|_{[k]}$  norms, and the complete convergence theory of Chapter 6, including the results of Section 6.5, extend to this more general context.

### 8.1.5 A More General Cauchy Point

In order to further illustrate the applicability of AA.1b and general criticality measures, we now consider a sufficient model decrease condition based on yet another way to define the Cauchy point. This is not the only motivation for the material of this

subsection, as it will also be used directly to analyse two practical algorithms in Sections 10.3.1 and 10.3.2. Recall that, in Section 6.3, our analysis was based on the definition of the Cauchy arc

$$x_k^C(t) \stackrel{\text{def}}{=} \{x \in \mathcal{B}_k \mid x = x_k - tg_k, t \geq 0\},$$

that is, the intersection of the steepest-descent direction with the trust region. We now investigate the model decrease that may be obtained if, instead of searching for the Cauchy point along the Cauchy arc, we merely search along a descent direction  $d_k$ , that is, a direction for which

$$\langle g_k, d_k \rangle < 0. \quad (8.1.34)$$

Specifically, we compute the *modified Cauchy point* by approximate or exact minimization along the modified Cauchy arc

$$\{x \in \mathcal{B}_k \mid x = x_k + td_k, t \geq 0\},$$

yielding the generalized form

$$x_k^{MC} = x_k + t_k^{MC} d_k. \quad (8.1.35)$$

Figure 8.1.2 illustrates the modified Cauchy arc in relation to the steepest-descent direction.

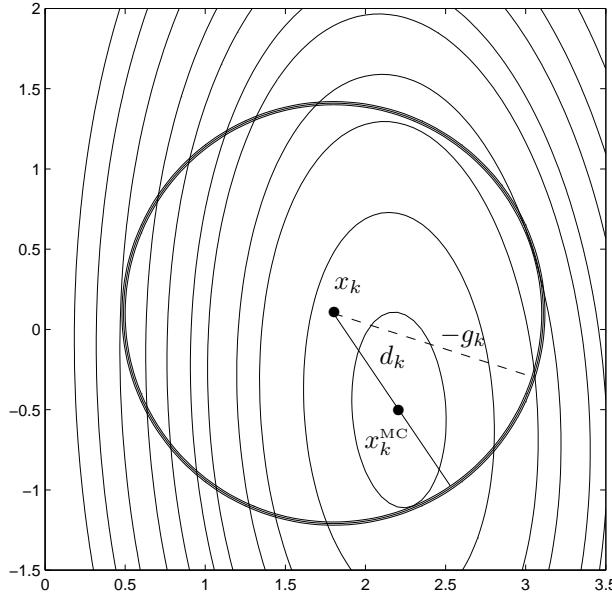


Figure 8.1.2: The steepest-descent direction from  $x_k$  (dashed) and the modified Cauchy arc  $d_k$  (solid) and modified Cauchy point  $x_k^{MC}$ .

The relevant part of Theorem 6.3.3 (p. 128), which specifies the model decrease we might expect at the modified Cauchy point, may then be rephrased as follows.

**Theorem 8.1.10** Suppose that AM.1 holds. Suppose furthermore that  $d_k$  satisfies (8.1.34) and that  $x_k^{\text{MC}} = x_k(j_c)$ , where

$$x_k(j) = x_k + \kappa_{\text{bck}}^j \frac{\Delta_k}{\|d_k\|_k} d_k, \quad (8.1.36)$$

$\kappa_{\text{bck}} \in (0, 1)$ , and  $j_c$  is the smallest nonnegative integer  $j$  for which

$$m_k(x_k(j)) \leq m_k(x_k) + \kappa_{\text{ubs}} \langle g_k, x_k(j) - x_k \rangle \quad (8.1.37)$$

for some  $\kappa_{\text{ubs}} \in (0, \frac{1}{2})$ . Then  $x_k^{\text{MC}}$  is well defined in the sense that  $j_c$  is finite and

$$m_k(x_k) - m_k(x_k^{\text{MC}}) \geq \kappa_{\text{dep}} \frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k} \min \left[ \frac{|\langle g_k, d_k \rangle|}{\beta_{\{k\}} \|d_k\|_k}, \Delta_k \right] \quad (8.1.38)$$

for some  $\kappa_{\text{dep}} \in (0, 1)$ .

**Proof.** The proof again parallels that of Theorem 6.3.3 (p. 128). We first consider the case where condition (8.1.37) is violated for some  $j$ , and therefore we have that

$$m_k(x_k + t_j d_k) > m_k(x_k) + \kappa_{\text{ubs}} t_j \langle g_k, d_k \rangle, \quad (8.1.39)$$

where

$$t_j = \frac{\kappa_{\text{bck}}^j \Delta_k}{\|d_k\|_k}. \quad (8.1.40)$$

We now use the mean value theorem for the left-hand side of (8.1.39) with a  $\xi_j$  belonging to the segment  $[x_k, x_k + t_j d_k]$ , and deduce that

$$m_k(x_k + t_j d_k) = m_k(x_k) + t_j \langle g_k, d_k \rangle + \frac{1}{2} t_j^2 \langle d_k, \nabla_{xx} m_k(\xi_j) d_k \rangle. \quad (8.1.41)$$

Now combining (8.1.39) and (8.1.41) and taking into account (8.1.14), the inclusion  $\xi_j \in \mathcal{B}_k$ , and (8.1.22), we obtain that

$$t_j > 2(1 - \kappa_{\text{ubs}}) \frac{|\langle g_k, d_k \rangle|}{\langle d_k, \nabla_{xx} m_k(\xi_j) d_k \rangle} \geq \frac{2(1 - \kappa_{\text{ubs}})}{\beta_{\{k\}} \|d_k\|_k} \frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k}.$$

As a consequence and because  $\kappa_{\text{bck}} < 1$ , there must be a first finite  $j_c$  such that

$$\kappa_{\text{bck}}^{j_c} \Delta_k < \frac{2(1 - \kappa_{\text{ubs}})}{\beta_{\{k\}} \|d_k\|_k} \frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k}, \quad (8.1.42)$$

for which we have that (8.1.37) holds. The modified Cauchy point  $x_k^{\text{MC}}$  is thus well defined and we obtain from (8.1.36) and (8.1.37) that

$$m_k(x_k) - m_k(x_k^{\text{MC}}) \geq -\kappa_{\text{ubs}} \langle g_k, x_k(j_c) - x_k \rangle = -\kappa_{\text{ubs}} \kappa_{\text{bck}}^{j_c} \Delta_k \frac{\langle g_k, d_k \rangle}{\|d_k\|_k}. \quad (8.1.43)$$

Now, if  $j_c \geq 1$  (that is,  $x_k^{\text{MC}}$  lies in the interior of the trust region) and since it is the smallest  $j$  that ensures (8.1.42), we may deduce from (8.1.40) and (8.1.42) that

$$\kappa_{\text{bck}}^{j_c} \Delta_k = \kappa_{\text{bck}} \kappa_{\text{bck}}^{j_c-1} \Delta_k \geq 2\kappa_{\text{bck}}(1 - \kappa_{\text{ubs}}) \frac{|\langle g_k, d_k \rangle|}{\beta_{\{k\}} \|d_k\|_k}, \quad (8.1.44)$$

which, together with (8.1.43), gives that

$$m_k(x_k) - m_k(x_k^{\text{MC}}) \geq 2\kappa_{\text{bck}}\kappa_{\text{ubs}}(1 - \kappa_{\text{ubs}}) \frac{|\langle g_k, d_k \rangle|^2}{\beta_{\{k\}} \|d_k\|_k^2}. \quad (8.1.45)$$

If, on the other hand,  $j_c = 0$ , that is,  $x_k^{\text{MC}}$  lies on the boundary of the trust region, we immediately deduce from (8.1.43) that

$$m_k(x_k) - m_k(x_k^{\text{MC}}) \geq \kappa_{\text{ubs}} \Delta_k \frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k}. \quad (8.1.46)$$

Combining (8.1.45) and (8.1.46) and using AN.1, we conclude that (8.1.38) holds with

$$\kappa_{\text{dep}} = \kappa_{\text{ubs}} \min [2\kappa_{\text{bck}}(1 - \kappa_{\text{ubs}}), 1] < 1. \quad (8.1.47)$$

□

The bound (8.1.38) is identical in form to AA.1b, so long as we choose the direction  $d_k$  to ensure that the quantity  $|\langle g_k, d_k \rangle|/\|d_k\|_k$  is a first-order criticality measure as defined in Section 8.1.1. This is the case, for instance, if we require that  $d_k$  is *gradient related* in the sense that

$$|\langle g_k, d_k \rangle| \geq \epsilon(x_k) \|g_k\| \|d_k\|, \quad (8.1.48)$$

where  $\epsilon(\cdot)$  is a continuous function from  $\mathbb{R}^n$  into the real interval  $(0, 1]$ , independent of  $d_k$ , such that  $\epsilon(x_k) = 0$  only if  $\|g_k\| = 0$ . In this case, we deduce that

$$\frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k} = \frac{\nu_k^{\text{MC}} |\langle g_k, d_k \rangle|}{\|d_k\|} \geq \nu_k^{\text{MC}} \epsilon(x_k) \|g_k\|, \quad (8.1.49)$$

where now

$$\nu_k^{\text{MC}} \stackrel{\text{def}}{=} \frac{\|d_k\|}{\|d_k\|_k}. \quad (8.1.50)$$

AN.1 and (8.1.17) then give us that (8.1.38) is nothing but AA.1b with

$$\pi(k, x_k) = \frac{\epsilon(x_k)}{\kappa_{\text{une}}} \|\nabla_x f(x_k)\|,$$

where  $\pi$  is again a criticality measure for the unconstrained minimization problem, because of the conditions we imposed on  $\epsilon(x)$ .

Condition (8.1.48) merely says that the angle between  $d_k$  and the steepest-descent direction must be strictly less than 90 degrees so long as  $x_k$  is not first-order critical. Note that the gradient-relatedness of  $d_k$  is independent of its norm  $\|d_k\|$ . Examples of how  $d_k$  might be computed will be given in Sections 10.3.1–10.3.3. Also note that we have used the uniform equivalence of norms (see AN.1) in the direction  $d_k$  instead of  $-g_k$ . This is to be expected because the Cauchy point is now defined along  $d_k$ . Finally, we could clearly use the condition that  $|\langle g_k, d_k \rangle| \geq \epsilon(x_k) \|g_k\|_{[k]} \|d_k\|_k$  instead of (8.1.48) if AN.1 does not hold, so long as one assumes that AN.1b and AN.3 hold.

A variant of Theorem 6.3.1 (p. 125) for quadratic models is also easy to establish using arguments similar to those we used in the last proof. As a consequence of this

development, we see that if  $d_k$  is gradient related in the sense of condition (8.1.48), a sufficient decrease property still holds if we choose the step  $s_k$  in such a way that it produces at least a fixed fraction of the model decrease obtained at the modified Cauchy point (where the search is performed along  $d_k$  instead of the steepest-descent direction). We may therefore apply the conclusions of Sections 8.1.2 and 8.1.3 (under AN.1), and deduce that *all convergence results of Sections 6.2 to 6.6 remain valid*, provided the assumptions stated therein<sup>120</sup> are satisfied. We also observe that the above discussion of convergence for general first-order measures indicates that AN.1 can be weakened to AN.1b and AN.3 for some of these results.

For future reference, we also derive a direct consequence of the proof of Theorem 8.1.10.

**Corollary 8.1.11** Suppose that AM.1 and AN.1 hold. Suppose furthermore that  $d_k$  satisfies (8.1.48) and that  $x_k^{\text{MC}} = x_k(j_c)$  is defined, as in Theorem 8.1.10, by (8.1.36) and (8.1.37). Then

$$t_k^{\text{MC}} \stackrel{\text{def}}{=} \frac{\kappa_{\text{bck}}^{j_c} \Delta_k}{\|d_k\|_k} \geq \kappa_{\text{dep}} \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\beta_k \|d_k\|}, \frac{\nu_k^{\text{MC}} \Delta_k}{\|d_k\|} \right]$$

for some  $\kappa_{\text{dep}} > 0$ .

**Proof.** We first note that AN.1, as it implies AN.3, also implies that

$$\beta_{\{k\}} \leq \kappa_{\text{une}}^2 \beta_k$$

because of (8.1.18). Given the definition of  $\nu_k^{\text{C}}$  in (8.1.50), the bound

$$t_k^{\text{MC}} = \frac{\kappa_{\text{bck}}^{j_c} \Delta_k}{\|d_k\|_k} \geq \kappa_{\text{dep}} \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\kappa_{\text{une}}^2 \beta_k \|d_k\|}, \frac{\nu_k^{\text{MC}} \Delta_k}{\|d_k\|} \right]$$

then immediately follows from (8.1.44), (8.1.47), and (8.1.49) if  $j_c \geq 1$ , and from the definition of  $t_k^{\text{MC}}$  and the fact that  $\kappa_{\text{dep}} \in (0, 1)$  if  $j_c = 0$ . The desired result then follows by redefining  $\kappa_{\text{dep}}$  as  $\kappa_{\text{dep}}/\kappa_{\text{une}}^2$ .  $\square$

Again, a similar result holds when the modified Cauchy point is determined by exact minimization on a quadratic model.

### 8.1.6 The Scaled Cauchy Point

We conclude this review of the convergence properties by examining a special case of the modified Cauchy point that arises naturally in the context of scaling. We now assume, as we did in Section 6.7, that a scaling of the space of variables is defined, at

---

<sup>120</sup>That is, AF.1–AF.3, AM.1–AM.4, AA.1b, AM.5–AM.6b, AA.2b, and AA.3. Note that AM.4 implies AM.4b when AN.1 or AN.1b holds, as we discussed on p. 252.

iteration  $k$ , by the norm  $\|\cdot\|_k$ . Then we may seek the direction  $d_k$  which corresponds to the direction of steepest descent in this new metric. This direction is, by definition, the minimizer of the problem

$$\inf_{d \neq 0} \frac{\langle g_k, d \rangle}{\|d\|_k} = -\sup_{d \neq 0} \frac{\langle g_k, d \rangle}{\|d\|_k}.$$

From this definition and the fact that equality can be attained in the Hölder inequality (8.1.12), we may deduce that

$$\langle g_k, d_k \rangle = -\|g_k\|_{[k]} \|d_k\|_k$$

and thus that

$$\frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k} = \|g_k\|_{[k]}. \quad (8.1.51)$$

If  $\|d\|_k = \|S_k^{-1}d\|$ , then

$$\sup_{d \neq 0} \frac{\langle g_k, d \rangle}{\|d\|_k} = \sup_{d \neq 0} \frac{\langle g_k, S_k S_k^{-1}d \rangle}{\|S_k^{-1}d\|} = \sup_{d \neq 0} \frac{\langle S_k^T g_k, S_k^{-1}d \rangle}{\|S_k^{-1}d\|} = \sup_{S_k x \neq 0} \frac{\langle S_k^T g_k, x \rangle}{\|x\|},$$

which gives that  $x = S_k^T g_k$  and  $d_k = -S_k S_k^T g_k$ . We also see that

$$\frac{|\langle g_k, d_k \rangle|}{\|d_k\|_k} = \|S_k^T g_k\| = \|g_k\|_{[k]}.$$

Because the steepest-descent direction provides the best local model improvement in the scaled space, we prefer to determine a modified Cauchy point in the sense of Section 8.1.5. What is the model decrease that can be achieved along the corresponding modified Cauchy arc? The answer is given in the following result.

**Theorem 8.1.12** Suppose that AM.1 holds. Suppose also that  $d_k$  is the steepest-descent direction corresponding to the norm  $\|\cdot\|_k$  and that  $x_k^{sc} = x_k(j_c)$ , where

$$x_k(j) = x_k + \kappa_{bck}^j \frac{\Delta_k}{\|d_k\|_k} d_k$$

and  $j_c$  is the smallest nonnegative integer  $j$  such that

$$m_k(x_k(j)) \leq m_k(x_k) + \kappa_{ubs} \langle g_k, x_k(j) - x_k \rangle.$$

Then  $x_k^{sc}$  is well defined in the sense that  $j_c$  is finite and

$$m_k(x_k) - m_k(x_k^{sc}) \geq \kappa_{dcp} \|g_k\|_{[k]} \min \left[ \frac{\|g_k\|_{[k]}}{\beta_{\{k\}}}, \Delta_k \right] \quad (8.1.52)$$

for some  $\kappa_{dcp} \in (0, 1)$ .

**Proof.** This is a direct application of (8.1.51) and Theorem 8.1.10.  $\square$

The point  $x_k^{\text{sc}}$  is called the *scaled Cauchy point*. We note that, even though in general  $d_k \neq -g_k$ , the lower bound (8.1.52) on the model decrease is identical to (8.1.21) in Theorem 8.1.9. Thus we see that both the usual Cauchy point  $x_k^C$  but also its scaled variant  $x_k^{\text{sc}}$  ensure that AA.1b holds for  $\pi_k = \|g_k\|_{[k]}$ . Finally, because a variant of Theorem 8.1.10 exists for the case where the Cauchy point is determined by exact minimization in the case of a quadratic model, the same is obviously true of Theorem 8.1.12.

## Notes and References for Section 8.1

The notion of a general criticality measure as an alternative to the gradient norm belongs to the folklore of the trust-region convergence theory; see, for instance, Conn, Gould, and Toint (1988a) or Burke, Moré, and Toraldo (1990) in the context of bound-constrained problems. Its formal statement may be found in Conn, Gould, Sartenaer, and Toint (1996a) for the more general context of linearly constrained problems. We have restrained ourselves to consider only the case of first-order measures, but it is also possible to investigate second-order ones, if one wishes to generalize AA.2b even further. For instance, we could consider

$$\varpi_k = \left( \max \left[ 0, -\lambda_{\min}[\nabla_{xx} m_k(x_k)] \right] \right)^\alpha$$

for some  $\alpha > 0$  and rephrase AA.2b to require that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} \max \left\{ \pi_k \min \left[ \frac{\pi_k}{\beta_k}, \Delta_k \right], \varpi_k \min[\varpi_k^2, \Delta_k^2, \kappa_{\text{lsd}}^2] \right\}$$

for some constant  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$ . We could also replace  $\min[\Delta_k, \kappa_{\text{lsd}}]$  by any function of  $\kappa(\Delta_k)$  such that

$$\lim_{\Delta \rightarrow 0} \frac{\kappa(\Delta)}{\Delta} \geq \epsilon$$

for some  $\epsilon > 0$ . A term of this nature could finally be included in the statement of AA.1b. However, the real applicability of such generalizations remains to be seen, and we have thus preferred not to develop our analysis any further.

Our exposition of convergence results for the case where we consider both primal and dual norms is inspired by a number of sources, including Conn et al. (1993) in the context of problems with convex constraints and Coleman and Li (1994) for the case where there are simple bounds on the variables. General norms were also considered in El-Hallabi and Tapia (1993) and El-Hallabi (1993) for the solution of systems of nonlinear equations. It is remarkable that AN.3 is only used twice, but at crucial points in the development: in the proof of (8.1.33) and in the generalized version of Theorem 6.6.4 (p. 154). We also note that only the effect of the norms in the directions  $g_k$  and  $s_k$  are ever mentioned, which leads to possibly still weaker versions of our assumptions.

Following the independent proposals of Carter (1987) and Toint (1981a, 1988) we may further pursue our aim of making the algorithmic concepts independent of the norms used, in that we may also free our convergence theory of its dependence on the norm of the objective function and model Hessian (as in AF.3, AM.4, and the definition of  $\beta_k$ ). This is done by replacing the quotients  $|\langle w, \nabla_{xx} h(x)w \rangle|/\|w\|_k^2$ , used in AM.4b, AM.5b, and AM.6b, by the more specific quantity

$$\omega_k(h, x, w) \stackrel{\text{def}}{=} \frac{h(x) - h(x + w) + \langle \nabla_x h(x), w \rangle}{\|w\|_k^2}$$

for any  $w \neq 0$  and any twice differentiable function  $h$  from  $\mathbb{R}^n$  into  $\mathbf{R}$ . If we assume that  $h$  is a quadratic function with Hessian  $H$ , then  $2\omega_k(h, x, w)$  is the Rayleigh quotient of  $H$  along the direction  $w$ . If one additionally chooses  $w$  to be along the eigenvector of  $H$  corresponding to its largest eigenvalue in absolute value, then  $2|\omega_k(h, x, w)| = \|H\|_2$ . For more general functions,  $\omega_k(h, x, w)$  measures the curvature of  $h$  along the direction  $w$ . Examining the results of Sections 6.2–6.6, we see that only

$$\omega_k(m_k, x_k, g_k), \quad \omega_k(m_k, x_k, s_k), \quad \omega_k(m_k, x_k, u_k), \quad \text{and } \omega_k(f, x_k, s_k)$$

play any role in our developments.<sup>121</sup> It is thus enough to replace AF.3 and AM.4 by the following weaker conditions.

**AF.3b** The curvature of the Hessian of the objective function along the step  $s_k$  is uniformly bounded; that is, there exists a constant  $\kappa_{\text{ufh}} > 0$  such that

$$\omega_k(f, x_k, s_k) \leq \kappa_{\text{ufh}}$$

for all  $k$ .

**AM.4c** The curvature of the Hessian of the model remains bounded within the trust region along the directions  $g_k$ ,  $s_k$ , and  $u_k$  (if applicable); that is,

$$\max \left[ \omega_k(m_k, x_k, g_k), \omega_k(m_k, x_k, s_k), \omega_k(m_k, x_k, u_k) \right] \leq \kappa_{\text{umh}} - 1$$

for all  $k$ , where  $\kappa_{\text{umh}} \geq 1$  is a constant independent of  $k$ .

We may therefore deduce that, if AF.3 and AM.4 are replaced by AF.3b and AM.4c, respectively, and if we assume that AN.1 and AN.3 hold, then all convergence results of Sections 6.2 to 6.6 are still valid.

The idea that a weaker definition of the Cauchy point is possible without altering the convergence properties of the basic algorithm first appeared in Shultz, Schnabel, and Byrd (1985).

## 8.2 Weakly Equivalent Norms

We now return to the first example of Section 6.7.2, where Algorithm BTR (p. 116) converges prematurely because the iteration-dependent norm defining the trust region “shrinks” too quickly compared to the  $\ell_2$  norm. As we have seen, the simplest course of action is then merely to impose AN.1, which ensures that the  $k$  and  $\ell_2$  norms always remain comparable. In the previous section, we adopted an attitude at the other extreme by replacing AN.1 by AN.1b and AN.3, but this did not ensure that limit points are first-order critical. However, a compromise position is also possible, where first-order criticality of limit points is preserved while AN.1 is weakened. For instance, instead of (6.7.10) (p. 166), we might consider using the trust region

$$\mathcal{B}_k = \{x \in \mathbf{R} \mid 4\|x - x_k\| \leq \|g_k\|\Delta_k\}. \quad (8.2.1)$$

<sup>121</sup>It is not surprising that these quantities specify the curvature of the objective and model along the same three directions that were of importance for scaling.

In other words, we consider the norm

$$\|x\|_k = \frac{4}{\|g_k\|} \|x\|,$$

which yields that  $\nu_k^C = \frac{1}{4}\|g_k\|$ . Since the objective function in our example is simply  $x^2$ , and if we keep  $\Delta_k = 1$  unchanged at successful iterations (as in Section 6.7.2), we deduce that  $\mathcal{B}_k = [\frac{1}{2}x_k, \frac{3}{2}x_k]$ . Hence

$$x_k = 2e \left( 1 - \sum_{i=1}^k \frac{1}{2^i} \right)$$

for  $k \geq 1$ , and thus  $\{x_k\}$  now converges to  $x_* = 0$ . The key point in obtaining the desired convergence in this example is that we have used a first-order criticality measure  $\pi_k = \|g_k\|$  in the definition of the norm to enforce that  $\|\cdot\|_k$  and the  $\ell_2$  norm remain comparable, at least if the current iterate is not critical.

Formally, we express this weakening in the lower bound of AN.1 as follows.

**AN.1c** There exist constants<sup>122</sup>  $\kappa_{cdn} > 0$  and  $\kappa_{une} > 0$  and a criticality measure  $\pi(\cdot)$  with values in  $[0, 1]$  such that

$$\kappa_{cdn} \pi(x_k) \|x\|_k \leq \|x\| \leq \kappa_{une} \|x\|_k$$

for all  $x \in \mathbb{R}^n$  and all  $k \geq 0$ .

We will refer to norms satisfying this assumption as *weakly equivalent*. In Figure 6.7.4 (p. 167), this amounts to making the diameter of the inner circle dependent on the criticality of the current iterate.

Because we have not modified the upper bound in AN.1c compared to AN.1, we immediately obtain that the right part of (6.7.11) (p. 168) remains true, as is the case in AN.1b. The only potential difficulty is therefore, as in Section 6.7, to ensure that AA.1 and AA.2 play their respective roles. We first verify that a bound of the type AA.1/AA.1b holds when we merely assume AN.1c. Returning to the basic properties of the Cauchy point, we deduce from substituting  $\pi_k$  for  $\|g_k\|$  in (6.3.4) (p. 125) and (6.3.19) (p. 128) that

$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2} \pi_k \min \left[ \frac{\pi_k}{\beta_k}, \nu_k^C \Delta_k \right]$$

and

$$m_k(x_k) - m_k(x_k^{AC}) \geq \kappa_{dep} \pi_k \min \left[ \frac{\pi_k}{\beta_k}, \nu_k^C \Delta_k \right].$$

Introducing AN.1c now implies that we have to use the bound  $\nu_k^C \geq \kappa_{cdn} \pi_k$  in the two inequalities above, instead of  $\nu_k \geq 1/\kappa_{une}$ . This yields that

$$\begin{aligned} m_k(x_k) - m_k(x_k^C) &\geq \min \left[ \frac{1}{2} \kappa_{cdn}, \kappa_{cdn} \kappa_{dep} \right] \pi_k^2 \min \left[ \frac{\pi_k}{\beta_k}, \Delta_k \right] \\ &\geq \min \left[ \frac{1}{2} \kappa_{cdn}, \kappa_{cdn} \kappa_{dep} \right] \bar{\pi}_k \min \left[ \frac{\bar{\pi}_k}{\beta_k}, \Delta_k \right], \end{aligned}$$

<sup>122</sup>“cdn” stands for “criticality-dependent norm”.

where  $\bar{\pi}_k \stackrel{\text{def}}{=} \pi_k^2 \leq \pi_k$  because  $\pi_k \in [0, 1]$ . Hence it is again very reasonable to require that the step  $s_k$  satisfy AA.1b, where the criticality measure now takes into account both the degree of criticality of  $x_k$  and its effect on  $\|\cdot\|_k$ .

This discussion is enough to guarantee convergence of an algorithm using norms that satisfy AN.1c to first-order critical points, that is, Theorem 6.4.6 (p. 137), but not to ensure that limit points will satisfy second-order conditions. For this to happen, we would have to reinterpret AA.2 to ensure that the  $\ell_2$  norm of the step to the eigenpoint also remains of the order of the trust-region radius. This is clearly possible, but goes beyond the level of generality that we feel is needed here. On the other hand, AN.1c also allows for an analysis of the convexity properties of the objective function in the cones  $\mathcal{K}_\epsilon$ , as described in Section 8.1.2.

## Notes and References for Section 8.2

The use of weak norm equivalence appears to be new but is inspired by the framework discussed by Coleman and Li (1994) for problems with simple bound constraints.

We could also transform the second example of Section 6.7.2 to relax AN.1 in the other direction. The idea here is that we must enforce a true reduction of the stepsize at unsuccessful iterations, at least asymptotically. Thus we would have to ensure that the bound  $\Delta_{k+1} \leq \gamma_1 \Delta_k$  eventually dominates the increase in  $\|s_k\|$  (compared to  $\|s_k\|_k$ ) that can result from an unfortunate choice of the sequence of norms. This could be achieved by imposing a condition like

$$\lim_{k \rightarrow \infty} \gamma_1^k \|x\|_k = 0 \quad \text{or} \quad \|x\|_k \geq \kappa_{\text{cdn}} \pi(x_k) \|x\| \quad (k \geq 0)$$

for all  $x \in \mathbb{R}^n$ .

## 8.3 Minimization in Infinite-Dimensional Spaces

Next, we consider extending our theory to cover infinite-dimensional spaces. Such an extension is justified by the practical observation that the infinite-dimensional properties of a problem tend to dominate in the solution of its discretized versions when the discretization is fine enough. Since we are interested in the efficient solution of such problems—for instance, in the context of optimal control—these infinite-dimensional properties are well worth investigating. If convergence can be established for an algorithm in the infinite-dimensional setting, its use on a discretized version of the problem is sometimes said to be “mesh-independent”, a property highly valued by practitioners. The purpose of this section is to show that Algorithm BTR (p. 116) is essentially mesh-independent in that sense.

We thus consider the problem of minimizing a functional  $f$  from a Banach space  $\mathcal{V}$  into  $\mathbb{R}$ . The differentiability assumption AF.1 must then be understood in the Fréchet sense (see, for instance, Ortega and Rheinboldt, 1970, p. 61). The gradient  $\nabla_x f(x)$  now no longer belongs to  $\mathcal{V}$ , but rather to its dual  $\mathcal{V}'$ , which is the space of all real bounded linear functionals on  $\mathcal{V}$ . In this context, we have to reinterpret the symbol

$\langle \cdot, \cdot \rangle$  as the dual pairing between  $\mathcal{V}$  and  $\mathcal{V}'$ , that is,

$$\langle y, x \rangle = y(x), \quad (8.3.1)$$

where  $x \in \mathcal{V}$  and  $y \in \mathcal{V}'$ , instead of interpreting this symbol as the inner product. Similarly, the norm  $\|\cdot\|$  now depends on  $\mathcal{V}$  and  $\mathcal{V}'$ , and convergence of iterates and gradients has to be interpreted in the strong topology, that is, the topology induced by the norm on each of these spaces. Thus our discussion of norms in the previous section now becomes crucial: if the vectors  $x \in \mathcal{V}$  are evaluated with the norm  $\|\cdot\|_{\mathcal{V}}$ , the dual quantities like the gradients must now be measured with the dual norm of  $\|\cdot\|_{\mathcal{V}}$ , which is  $\|\cdot\|_{\mathcal{V}'}$  (see also (8.1.11)). For instance, Theorem 6.4.6 (p. 137) must be rephrased as

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\|_{\mathcal{V}'} = 0$$

(see also (8.1.33)). Furthermore, the Hessian of  $f$ ,  $\nabla_{xx} f$ , is now a linear operator from  $\mathcal{V}$  into  $\mathcal{V}'$ . Finally, notice that, if  $s$  and  $d$  belong to  $\mathcal{V}$  and  $H$  is a linear operator from  $\mathcal{V}$  to  $\mathcal{V}'$ , then

$$|\langle s, Hd \rangle| \leq \|s\|_{\mathcal{V}} \|Hd\|_{\mathcal{V}'} \leq \|s\|_{\mathcal{V}} \|H\|_{\mathcal{V}, \mathcal{V}'} \|d\|_{\mathcal{V}},$$

where

$$\|H\|_{\mathcal{V}, \mathcal{V}'} \stackrel{\text{def}}{=} \sup_{\|x\|_{\mathcal{V}}=1} \|Hx\|_{\mathcal{V}'}$$

is the norm induced by the  $\mathcal{V}$  and  $\mathcal{V}'$  norms on the space of linear functionals from  $\mathcal{V}$  to  $\mathcal{V}'$  (see (8.1.13)). In particular, this means that AF.3 has to hold in the norm  $\|\cdot\|_{\mathcal{V}, \mathcal{V}'}$ .

### 8.3.1 Hilbert Spaces

Fortunately, there is a case where the added generality does not result in significant additional complication to our convergence theory. If we restrict ourselves to Hilbert spaces, then this theory can be extended without much trouble. The key is that, if  $\mathcal{V}$  is a Hilbert space, then Riesz's representation theorem (see, for instance, Yosida, 1970, p. 90) implies that  $\mathcal{V}'$  can be identified with  $\mathcal{V}$ . In this case, we may retain the interpretation of  $\langle \cdot, \cdot \rangle$  to mean the inner product, and  $\|\cdot\|_{\mathcal{V}} = \|\cdot\|_{\mathcal{V}'}$ . As a consequence, nearly all the arguments developed in our global convergence analysis of Algorithm BTR (p. 116) remain valid without any modification. The first exception is that we have to abandon the notion of model minimizer  $x_k^M$  (see p. 131), because (6.3.28) may not have a solution. For consider the example

$$\min_{x \in L^2[0,1]} - \int_0^1 t x^2(t) dt \text{ subject to } \|x(t)\|_{L^2[0,1]} \leq 1,$$

where  $L^2[0, 1]$  is the Hilbert space of real-valued square-integrable functions on the interval  $[0, 1]$ . Note that this is a quadratic minimization problem subject to a norm constraint, and thus that it has the same form as the subproblem we considered in detail in Chapter 7 (with  $\Delta_k = 1$ ). To analyse this example, we first note that

$$-\int_0^1 t x^2(t) dt \geq - \left[ \max_{t \in [0,1]} t \right] \int_0^1 x^2(t) dt \geq -1$$

and that

$$x_\alpha(t) = \sqrt{2\alpha+1} t^\alpha \in L^2[0,1] \text{ and } \|\sqrt{2\alpha+1} t^\alpha\|_{L^2[0,1]} = 1$$

for all  $\alpha > -\frac{1}{2}$ . Evaluating the objective for this  $x_\alpha(t)$ , we find that

$$-\int_0^1 t x_\alpha^2(t) dt = -(2\alpha+1) \int_0^1 t^{2\alpha+1} dt = -\frac{2\alpha+1}{2\alpha+2}.$$

Hence we may pick any sequence  $\{\alpha_i\}$  converging to plus infinity, and we obtain a sequence of  $x_{\alpha_i}(t)$  such that the value of the objective functional converges to the lower bound  $-1$ . This lower bound is thus the infimum of the problem, but is only attained in the limit. Furthermore, we now show that no other functional on the unit sphere will attain the lower bound. Indeed, it is clear that this lower bound cannot be achieved for any  $x$  such that  $\|x\|_{L^2[0,1]} = 0$ . Consider now an  $x$  such that  $0 < \|x\|_{L^2[0,1]} \leq 1$ . Then we have that

$$\int_0^1 t x^2(t) dt = \int_0^1 x^2(t) dt - \int_0^1 (1-t)x^2(t) dt \leq 1 - \int_0^1 (1-t)x^2(t) dt. \quad (8.3.2)$$

Furthermore, since  $\|x\|_{L^2[0,1]} > 0$ , there must exist an  $\epsilon > 0$  and a subset  $\mathcal{Z} \subset [0,1]$  with Lebesgue measure  $\mu(\mathcal{Z}) > 0$  such that  $|x(t)| \geq \epsilon$  for all  $t \in \mathcal{Z}$ . Hence

$$\int_0^1 (1-t)x^2(t) dt \geq \epsilon^2 \int_{\mathcal{Z}} (1-t) dt \geq \epsilon^2 \int_{1-\mu(\mathcal{Z})}^1 (1-t) dt = \frac{1}{2}\epsilon^2 \mu(\mathcal{Z})^2 > 0,$$

which, together with (8.3.2), implies that

$$\int_0^1 t x^2(t) dt \leq 1 - \frac{1}{2}\epsilon^2 \mu(\mathcal{Z}) < 1,$$

and the lower bound cannot be attained. In fact, the deeper reason why difficulties could arise with the model minimizer is that the unit sphere may not be compact in infinite-dimensional spaces, which explains why the infimum may not be attained. However, if  $x_k^M$  may fail to exist, we can still rely on the notion of optimal decrease by replacing  $m_k(x_k^M)$  (in, for instance, Theorem 6.3.5 [p. 132]) by  $\inf_{x \in \mathcal{B}_k} m_k(x)$ . Or, more simply, we can still use the notion of a Cauchy point. Note that the existence of this point is ensured because it results from a minimization process over an (obviously finite-dimensional) segment, a much simpler problem. For the same reason, we have to redefine  $\beta_k$  to be

$$\beta_k \stackrel{\text{def}}{=} 1 + \sup_{x \in \mathcal{B}_k} \|\nabla_{xx} m_k(x)\|_{\mathcal{V}, \mathcal{V}'},$$

because the maximum in (6.3.1) (p. 124) may not be attained. The second slight modification of our theory for Hilbert spaces is that we have to take into account that the spectrum of  $\nabla_{xx} f(x)$  is no longer a finite discrete set of numbers. As a consequence, the notion of “minimum eigenvalue of the Hessian”  $\lambda_{\min}[H]$ , which is used in Theorem 6.5.2 (p. 141) and thereafter, may no longer be well defined. But the problem can be easily circumvented by redefining

$$\lambda_{\min}[H] \stackrel{\text{def}}{=} \inf_{d \in \mathcal{V}, d \neq 0} \frac{\langle d, Hd \rangle}{\langle d, d \rangle} \quad (8.3.3)$$

for each linear operator  $H$  from  $\mathcal{V}$  into  $\mathcal{V}' = \mathcal{V}$ . Furthermore, the “min” in (6.5.3) (p. 141) should be replaced by an “inf”, as we just explained for the model minimizer. It also follows from the redefinition (8.3.3) that the eigenpoint is well defined and the convergence theory for second-order critical points is applicable to our more general context. The only result that does not extend is Theorem 6.6.5 (p. 155), because its proof uses a compactness argument that is in general no longer true in Hilbert spaces.

### 8.3.2 Banach Spaces

We will not reexamine the contents of Chapter 6 in detail for the more general case of minimization over a Banach space. Rather, we will attempt to point out where the difficulties are and how they can be solved.

The first problem is the definition of the Cauchy point. We note that, in this more general context, the negative gradient  $-\nabla_x f(x)$  does *not* specify the direction of steepest descent, which is now given by any direction  $d \in \mathcal{V}$  such that

$$\langle d, \nabla_x f(x) \rangle = -\|d\|_{\mathcal{V}} \|\nabla_x f(x)\|_{\mathcal{V}'}.$$

We could thus think of defining a new Cauchy point concept by finding the point that (approximately) minimizes the model on the intersection of the positive multiples of  $d$  and the trust region, leading to a reformulation of AA.1. However, the real use of the gradient norm in AA.1 is *not* related to the fact that, in finite dimensions, the negative gradient and steepest-descent direction coincide, but rather with the fact that the first-order necessary conditions for optimality are written in terms of the gradient (see Theorem 3.2.1 [p. 38]). In other words, it is crucial that any reformulation of AA.1 ensure that progress can be made on the model as long as  $x_k$  is not a first-order critical point. Since the first-order necessary condition is also  $\nabla_x f(x) = 0$  in infinite-dimensional Banach spaces, we thus wish to reformulate our Cauchy decrease condition in terms of the gradient. One way to ensure that this is possible is to make the following additional assumption.

**AF.6** The gradient of  $f$ ,  $\nabla_x f(x)$ , belongs to  $\mathcal{V}$  for all  $x \in \mathcal{V}$ . Furthermore,  $\nabla_x f$  is uniformly continuous from  $\mathcal{V}$  to  $\mathcal{V}$ .

This assumption is not as restrictive as it might seem at first glance. Of course, it will not be satisfied by all functionals  $f$  satisfying the infinite-dimensional version of AF.1, but it may still allow for a number of interesting cases. For instance, if we choose  $\mathcal{V} = L^p(\Omega)$ , the Lebesgue space of real-valued  $p$ -integrable ( $2 \leq p < \infty$ )<sup>123</sup> functions over a domain  $\Omega$  of  $\mathbb{R}^n$  with positive and finite Lebesgue measure  $\mu(\Omega)$ , the first part

---

<sup>123</sup>The case  $p = \infty$  is slightly more complicated, because the dual of  $L^\infty(\Omega)$ ,  $L^\infty(\Omega)'$  is not identical to  $L^1(\Omega)$ . However, the framework discussed here may still be extended to this case because the operator that associates  $\langle \cdot, w \rangle \in L^\infty(\Omega)'$  to each  $w \in L^1(\Omega)$  is a linear norm-preserving injection, and hence we may interpret  $L^1(\Omega)$  as a subspace of  $L^\infty(\Omega)'$ . However, we then need  $\nabla_x f$  to map  $L^\infty(\Omega)$  continuously into  $L^1(\Omega)$ . Note that  $\nabla_x f$  is then continuous because the norms  $\|\cdot\|_{L^1(\Omega)} = \|\cdot\|_{L^\infty(\Omega)'}$  coincide on  $L^1(\Omega)$ .

of AF.6 only requires that the gradient of  $f$  must lie in  $L^p(\Omega)$ , which is a subset of  $L^{p'}(\Omega)$ , where

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

This is often acceptable as, in many applications, an even stronger assumption can be made, namely, that  $\nabla_x f(x) \in L^\infty(\Omega)$  (which means that the gradients are essentially bounded on  $\Omega$ ).

Thus, if AF.6 holds, then  $\|\nabla_x f(x)\|_{\mathcal{V}'}$  is well defined. But this is not enough in general to guarantee that the negative gradient  $-\nabla_x f(x)$  provides a descent direction in  $\mathcal{V}$ . In order to obtain this very desirable property, we make a further assumption.

**AN.2** For every vector  $x \in \{x \in \mathcal{V} \mid f(x) \leq f(x_0)\}$ , one has that

$$\langle \nabla_x f(x), \nabla_x f(x) \rangle \geq \phi(\|\nabla_x f(x)\|_{\mathcal{V}'}) \|\nabla_x f(x)\|_{\mathcal{V}}$$

for some continuous monotonically increasing real function  $\phi$  from  $[0, \infty]$  to itself, independent of  $x$  and such that  $\phi(0) = 0$  and  $\phi(t) > 0$  for  $t > 0$ .

Note that, because of AF.6, the first argument of the dual pairing can be seen as a vector of  $\mathcal{V}$  and the second as a vector of  $\mathcal{V}'$ , as required by (8.3.1). We then obtain from AN.2 that

$$\langle -g_k, g_k \rangle \leq -\phi(\|g_k\|_{\mathcal{V}'}) \|g_k\|_{\mathcal{V}},$$

as desired, and the negative gradient then provides a direction for the determination of the Cauchy point that is thus also well defined. This allows us to show that

$$m_k(x_k) - m_k(x_k^C) \geq \kappa_{\text{dep}} \frac{|\langle g_k, g_k \rangle|}{\|g_k\|_{\mathcal{V}}} \min \left[ \frac{|\langle g_k, g_k \rangle|}{\beta_k \|g_k\|_{\mathcal{V}}}, \Delta_k \right].$$

We may then use AN.2 again to obtain that

$$m_k(x_k) - m_k(x_k^C) \geq \kappa_{\text{dep}} \phi(\|g_k\|_{\mathcal{V}'}) \min \left[ \frac{\phi(\|g_k\|_{\mathcal{V}'})}{\beta_k}, \Delta_k \right],$$

leading to the now familiar form of AA.1, with the role of  $\|g_k\|$  now being played by  $\phi(\|g_k\|_{\mathcal{V}'})$ . This is perfectly adequate for our theory, because the complete convergence analysis can be carried out with the criticality measure  $\pi(x_k) = \phi(\|g_k\|_{\mathcal{V}'})$  (see Section 8.1).

Again, AN.2 may seem stronger than it really is. As above, consider the choice  $\mathcal{V} = L^p(\Omega)$ ,  $2 \leq p < \infty$ , and assume that  $\|g\|_{L^p(\Omega)} \leq \kappa_{\text{ubg}}$  for some  $\kappa_{\text{ubg}} > 0$  independent of  $g$ .<sup>124</sup> The dual pairing is then

$$\langle y(t), x(t) \rangle = \int_{\Omega} x(t)y(t) dt$$

and the properties of the  $L^p$  norms then imply that

$$\langle g, g \rangle = \int_{\Omega} g^2(t) dt = \|g(t)\|_{L^2(\Omega)}^2 \geq \frac{1}{\mu(\Omega)^{2(\frac{1}{p'} - \frac{1}{2})}} \|g\|_{L^{p'}(\Omega)}^2 \geq \phi(\|g\|_{L^{p'}(\Omega)}) \|g\|_{L^p(\Omega)}$$

<sup>124</sup>A priori bounds of this type can indeed be obtained in applications, especially when the level set  $\{x \in \mathcal{V} \mid f(x) \leq f(x_0)\}$  is known to be bounded.

for all  $g \in L^p(\Omega) \cap L^{p'}(\Omega)$ , where

$$\phi(t) = \frac{t^2}{\kappa_{\text{ubg}} \mu(\Omega)^{2(\frac{1}{p'} - \frac{1}{2})}}.$$

AN.2 therefore automatically holds in this important case. The same is true in the more general case of a “weak” Gelfand (evolution) triple,<sup>125</sup> which is defined by assuming that  $\mathcal{V}$  is densely and continuously embedded in some Hilbert space  $\mathcal{H}$ . Classical functional analysis results then imply that  $\mathcal{H} = \mathcal{H}'$  is continuously embedded in  $\mathcal{V}'$  by the mapping from  $v \in \mathcal{H}$  to  $\langle v, \cdot \rangle \in \mathcal{V}'$ , where  $\langle \cdot, \cdot \rangle$  represents the inner product on  $\mathcal{H}$ . An argument similar to that for  $L^p$  spaces then applies, provided one assumes that  $\|g\|_{\mathcal{V}} \leq \kappa_{\text{ubg}}$ , and AN.2 again holds. This covers, for instance, the case where  $\mathcal{V}$  is a Sobolev space  $W^{k,p}(\Omega)$ ,  $p > 2$ , and  $\mathcal{H}$  is  $W^{k,2}(\Omega)$ . A trust-region algorithm in this framework might thus be applied to a class of distributed control problems in partial differential equations or to the solution of nonlinear partial differential equations.

Given AF.6 and AN.2, the complete convergence theory for Algorithm BTR (p. 116), except Theorem 6.6.5 (p. 155), may then be extended to Banach spaces—a very general result. We conclude this section with the observation that scaling can also be taken into account in infinite-dimensional spaces (as in Section 6.7), provided that the infinite-dimensional analog of AN.1 holds. In our  $L^p(\Omega)$  context, we could, for instance, define the scaled norm  $\|x(t)\|_k$  as  $\|w_k(t)x(t)\|_{L^p(\Omega)}$ , where  $w_k(t)$  is a positive real-valued function in  $L^\infty(\Omega)$  such that both  $\|w_k(t)\|_{L^\infty(\Omega)}$  and  $\|w_k(t)^{-1}\|_{L^\infty(\Omega)}$  are uniformly bounded with respect to  $k$ .

## Notes and References for Section 8.3

The importance of considering the original infinite-dimensional problem in the solution of discretized instances is argued, for instance, by Kelley and Sachs (1987), Allgower et al. (1986), Deuflhard and Potra (1992), Heinkenschloss (1993), and Ulbrich and Ulbrich (1997). The extension of the results on global convergence to first-order critical points in the context of Hilbert spaces is presented in Toint (1988). This reference considers the more general problem of optimization subject to convex constraints. An extension to Banach spaces is proposed in Kelley and Sachs (1999) in the context of parabolic boundary control problems, which requires handling pointwise bounds on the (control) variables. Our presentation is inspired by Ulbrich, Ulbrich, and Heinkenschloss (1999), which contains more results in the context of minimization subject to pointwise bounds in  $L^p$  spaces. It also contains an elaboration on first-order and second-order criticality conditions in infinite-dimensional spaces, a subject already studied in, for example, Maurer and Zowe (1979). We refer the reader to Wloka (1987) for further details on abstract spaces and Gelfand triples.

The theory of this section is relevant not only for a class of unconstrained optimal control applications (see, for instance, Fukushima and Yamamoto, 1986, Coleman and Liao, 1995, Liao, 1995, 1997, for suitable trust-region algorithms), but also to more general problems whose

---

<sup>125</sup>What we describe here is not quite a Gelfand or evolution triple, for which  $\mathcal{V}$  is also assumed to be reflexive, and thus  $\mathcal{H}$  to be dense in  $\mathcal{V}'$ .

objective function involves the solution of some infinite-dimensional problem, for example, the solution of ordinary or partial differential equations (see Zhu and Brown, 1987, Cheng and Sykulski, 1996, and Lukšan and Vlček, 1996, for examples).

## 8.4 Using Approximate Derivatives

### 8.4.1 Concepts and Assumptions

We continue examining possible extensions of our basic trust-region algorithm and devote this section to cases where the model is *approximate* in the sense that we will relax the requirement that the objective and model gradients coincide (as in AM.3) and the condition that the Hessians of the model remain uniformly bounded (as in AM.4).

There are many reasons why one may wish to build models whose derivatives differ from those of the objective function. Either these derivatives may be unavailable, or they could be obtained only by some approximate calculation, such as by finite differences. This may not only be the case for the gradient, but also for the Hessian. In particular, we may wish to adapt our trust-region algorithm to handle quasi-Newton approximations.

#### 8.4.1.1 Inexact Gradients

Of course, if we wish to relax AM.3 and AM.4, we must indicate precisely in what sense. Consider AM.3 first, that is, the relation between the gradient of the model and that of the objective function. We shall require the following condition on the error

$$v_k \stackrel{\text{def}}{=} \nabla_x f(x_k) - g_k. \quad (8.4.1)$$

**AM.3b** There exists a constant<sup>126</sup>  $\kappa_{\text{egg}}$  such that

$$\|v_k\| \leq \kappa_{\text{egg}} \|g_k\|$$

for all  $k$  and

$$0 \leq \kappa_{\text{egg}} < \frac{\kappa_{\text{mdc}}(1 - \eta_2)}{\kappa_{\text{une}}}.$$

This condition states that the *absolute* error on the gradient must go to zero when the gradient of the model itself converges to zero. We state this simple result in slightly different form for future reference. (The former statement can be recovered using  $\|v_k\| \leq \kappa_{\text{egg}} \|g_k\|$ .)

---

<sup>126</sup>“egg” stands for “error on the gradient relative to the model’s g

**Lemma 8.4.1** Suppose that AM.3b holds and that there exists a subsequence of iterates  $\{x_{k_i}\}$  such that

$$\lim_{i \rightarrow \infty} g_{k_i} = 0. \quad (8.4.2)$$

Then

$$\lim_{i \rightarrow \infty} \nabla_x f(x_{k_i}) = 0. \quad (8.4.3)$$

Conversely, (8.4.3) implies (8.4.2).

**Proof.** The first part of the result immediately follows from the bound

$$\|\nabla_x f(x_{k_i})\| \leq \|v_{k_i}\| + \|g_{k_i}\| \leq (\kappa_{\text{egg}} + 1)\|g_{k_i}\|,$$

where we used (8.4.1), the triangle inequality, and AM.3b. The second part similarly uses the inequality

$$\|g_{k_i}\| \leq \|\nabla_x f(x_{k_i})\| + \|v_{k_i}\| \leq \|\nabla_x f(x_{k_i})\| + \kappa_{\text{egg}}\|g_{k_i}\|,$$

itself resulting from (8.4.1), the triangle inequality, and AM.3b. Moreover, AM.3b and the bound  $\kappa_{\text{mdc}} < 1$  imply that  $\kappa_{\text{egg}} < 1$ . We may then deduce that

$$\|g_{k_i}\| \leq \frac{\|\nabla_x f(x_{k_i})\|}{1 - \kappa_{\text{egg}}},$$

from which the desired property immediately follows.  $\square$

Hence, first-order critical points of the models correspond to first-order critical points of the objective function. Also note that the upper bound on  $\kappa_{\text{egg}}$  implies that  $\kappa_{\text{mdc}}$  is known. This is not a real drawback, since Section 6.3 shows that this constant is either  $1/2\kappa_{\text{une}}$  for quadratic models<sup>127</sup> or otherwise depends on constants from the technique used to determine the Cauchy point ( $\kappa_{\text{bck}}$  and  $\kappa_{\text{ubs}}$ ), whose values are known.

#### 8.4.1.2 Weaker Assumptions on the Model Hessians

Consider now our assumption of the Hessians of the models, AM.4. One of the main motivations for relaxing this assumption is to allow for quasi-Newton approximations of the Hessian. These techniques use the “quasi-Newton” or “secant” equation

$$H_{k+1}s_k = g_{k+1} - g_k \stackrel{\text{def}}{=} y_k$$

to update an existing approximation  $H_k$  to obtain a better one,  $H_{k+1}$ , typically by projecting  $H_k$  onto the subspace of symmetric matrices satisfying the secant equation with respect to some weighted Frobenius norm. The Broyden–Fletcher–Goldfarb–Shanno (BFGS) update

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{\langle s_k, H_k s_k \rangle} + \frac{y_k y_k^T}{\langle y_k, s_k \rangle} \quad (8.4.4)$$

---

<sup>127</sup>Implying in turn that  $\kappa_{\text{egg}}$  should be chosen in the interval  $[0, \frac{1}{2}(1 - \eta_2))$  if  $\|\cdot\|_k = \|\cdot\|$  for all  $k$ .

and the symmetric rank 1 (SR1) update

$$H_{k+1} = H_k + \frac{(y_k - H_k s_k)(y_k - H_k s_k)^T}{\langle y_k - H_k s_k, s_k \rangle} \quad (8.4.5)$$

are two well-known and widely used formulae from this class. The first is well defined provided the inner product  $\langle y_k, s_k \rangle$  is strictly positive, while the second requires  $\langle y_k - H_k s_k, s_k \rangle$  to be nonzero.

Quasi-Newton updating formulae (such as (8.4.4) or (8.4.5)) are typically applied in three useful contexts. The first and best known is to recur the Hessian approximation for the objective function from iteration to iteration, in either dense or factorized form. The second is to apply the updating formulae on each of the elements of the objective function, in the very frequent case where it is partially separable.<sup>128</sup> This technique, known as “partitioned updating”, has proved to be remarkably well suited to cases where the number of problem variables is large.<sup>129</sup> The third main application of quasi-Newton updates exploits the fact that they consist of low (1 or 2) rank corrections, and thus that a limited number of these corrections may be kept in the computer’s memory for large problems. The idea of “limited-memory quasi-Newton” methods is then to define the current Hessian approximation by applying these corrections to a well-chosen, easily invertible matrix. Since quasi-Newton methods are not our main interest, we will not develop this subject further, but the reader should be aware of the vast literature devoted to this topic.<sup>130</sup> It is enough for our purposes to consider bounds on the growth of  $\|H_k\|$  as the iterations proceed. Consider the BFGS update (8.4.4) first and assume that the objective function is convex. Then the mean value theorem implies that

$$y_k = G_k s_k,$$

where

$$G_k = \int_0^1 \nabla_{xx} f(x_k + ts_k) dt$$

is positive definite. The hereditary positive definiteness property of the BFGS update (see Dennis and Schnabel, 1983, p. 199) then guarantees that  $H_k$  is also positive definite<sup>131</sup> and hence (8.4.4) can be rewritten as

$$H_{k+1} = H_k^{\frac{1}{2}} \left( I - \frac{u_k u_k^T}{\|u_k\|^2} \right) H_k^{\frac{1}{2}} + G_k^{\frac{1}{2}} \left( \frac{w_k w_k^T}{\|w_k\|^2} \right) G_k^{\frac{1}{2}}, \quad (8.4.6)$$

where

$$u_k = H_k^{\frac{1}{2}} s_k \text{ and } w_k = G_k^{\frac{1}{2}} s_k.$$

<sup>128</sup>We review this structure in more detail in Section 10.2, where a formal definition of partially separable functions and their elements can be found. At this stage, it is enough to say that functions expressed as a sum of terms (the elements), each involving a small subset of the problem’s variables, are partially separable. In this latter case, the Hessian of each term is approximated by a separate (low-rank) quasi-Newton matrix.

<sup>129</sup>See also Chen, Deng, and Zhang (1998).

<sup>130</sup>See Dennis and Schnabel (1983), Chapter 9, for an introduction to the subject.

<sup>131</sup>Provided the initial approximation  $H_0$  is positive definite.

We now note that the terms within brackets in the right-hand side of (8.4.6) are orthogonal projections and thus of norm at most 1, which implies that

$$\|H_{k+1}\| \leq \|H_k^{\frac{1}{2}}\|^2 + \|G_k^{\frac{1}{2}}\|^2 = \|H_k\| + \|G_k\|.$$

Recalling AF.3, we see that  $\|G_k\| \leq \kappa_{\text{ufh}}$ , and thus that

$$\|H_{k+1}\| \leq \|H_k\| + \kappa_{\text{ufh}}. \quad (8.4.7)$$

A similar bound may be obtained for the SR1 update (8.4.5) without any convexity hypothesis, if one assumes that the update is performed only when

$$|\langle y_k - H_k s_k, s_k \rangle| \geq \kappa_{\text{sr1}} \|y_k - H_k s_k\|^2$$

for some constant  $\kappa_{\text{sr1}} > 0$ . For we immediately obtain from (8.4.5) that

$$\|H_{k+1}\| \leq \|H_k\| + \kappa_{\text{sr1}}^{-1}. \quad (8.4.8)$$

Our relaxed assumption on the growth of the model Hessians takes the bounds (8.4.7) and (8.4.8) into account and is formulated in terms of the bound on the curvature of the model  $\beta_k$  in (6.3.1) (p. 124).

**AM.4d** We assume that

$$\sum_{k=0}^{\infty} \frac{1}{\varphi_k} = \infty,$$

where  $\varphi_k$  is defined to be

$$\varphi_k = 1 + \max_{j=0,\dots,k} \max_{x \in \mathcal{B}_j} \|\nabla_{xx} m_j(x)\|. \quad (8.4.9)$$

We immediately observe that

$$\varphi_k = 1 + \max_{j=0,\dots,k} [\beta_j - 1],$$

where  $\beta_j$  was defined in (6.3.1) (p. 124), and hence that

$$\beta_k \leq \varphi_k \quad (8.4.10)$$

for every  $k$ . Moreover, the definition of  $\varphi_k$  implies that the sequence  $\{\varphi_k\}$  is nondecreasing.

For the BFGS update, (8.4.7) implies that

$$\varphi_k \leq \|H_0\| + \kappa_{\text{ufh}} k,$$

and thus AM.4d holds because the harmonic series is divergent. Similarly, for the SR1 update, (8.4.8) implies that

$$\varphi_k \leq \|H_0\| + \kappa_{\text{sr1}}^{-1} k$$

and AM.4d again holds for the same reason. Moreover, it is not too difficult to verify that AM.4d cannot be weakened without impairing the convergence properties of our

algorithm. Indeed, consider the problem of minimizing the function of a single variable  $f(x) = x$  using Algorithm BTR (p. 116), choosing the model at iteration  $k$  defined to be

$$m_k(x_k + s) = x_k + s + \varphi_k \frac{s^2}{2}.$$

The model minimizer is then given by

$$x_k^M = x_k - \frac{1}{\varphi_k}.$$

If the initial trust-region radius is chosen large enough to contain  $x_0^M$ , all iterations are then successful and the trust-region radii remain inactive throughout the calculation. Furthermore, we obtain that

$$x_k = x_0 - \sum_{i=0}^{k-1} \frac{1}{\varphi_i}.$$

Hence, if AM.4d is violated, this sum is finite and the algorithm cannot converge to the minimizer (at minus infinity), because the increasing curvature of the model forces shorter and shorter steps, in effect preventing the iterates from leaving a fixed ball centered at the starting point  $x_0$ . As we will see further, AM.4d is not quite sufficient to obtain all the global convergence properties that we may wish, but it is enough to start our analysis.

Interestingly, we need a further assumption on our algorithm in order to prove the desired global convergence result. It is reminiscent of the upper bound in AA.3 and specifies the mechanism for updating the trust-region radius as follows.

**AA.4** If  $\rho_k \geq \eta_2$ , then  $\Delta_{k+1} \leq \gamma_5 \Delta_k$  for some  $\gamma_5 \geq 1$ .

As the attentive reader may verify in the remainder of this section, this additional condition on Algorithm BTR is only needed whenever  $\varphi_k$  is allowed to tend to infinity. The assumptions AM.4d and AA.4 together guarantee that the rates of growth of the trust-region radius and the model Hessian are not independent of each other.

We conclude this discussion of the assumptions needed to establish the convergence of methods that use approximate derivative information in the model by reminding the reader that these assumptions generalize those we made in Chapter 6. For if exact gradients are used, this is subsumed by AM.3b with  $\kappa_{\text{egg}} = 0$ . Moreover, bounded Hessian approximations, as specified by AM.4, are also a special case of AM.4d, because AM.4 and (8.4.9) together ensure that  $\varphi_k$  is bounded and thus that AM.4d also holds. This is not the case for AA.4, which effectively restricts, albeit in a practically reasonable way, the range of possible realizations of Algorithm BTR.

We also note that the assumption on the use of inexact gradients, AM.3b, is independent of AM.4d. For instance, our framework covers the case where exact gradients are used in conjunction with quasi-Newton approximations of the Hessian. Similarly, it also covers the case where inexact gradients are used with AM.3b together with bounded approximations of the Hessian.

## Notes and References for Subsection 8.4.1

Software for unconstrained problems incorporating quasi-Newton updating to obtain approximations of second derivatives is too abundant to review thoroughly. The first program to be cast in a trust-region framework appears to be that of Powell (1970a). Other contributions include UNCMIN (see Schnabel, Koontz, and Weiss (1985), NL2SOL (see Gay, 1981, and Dennis, Gay, and Welsch, 1981), the routines of Gay (1983), VE08 (see Toint, 1983b), VE10 (see Toint, 1987b), and LANCELOT (see Conn, Gould, and Toint, 1992b), the latter three making use of partitioned updating for large structured problems. Many of these packages also allow the approximation of gradients by finite differences (see Section 8.4.3.1). One should also note that quasi-Newton approaches are not restricted to quadratic models: the classes of conic and tensor models mentioned on p. 162 also fall into that category. Finally, we mention that the question of sizing quasi-Newton Hessian approximations (that is, multiplying them by a scalar factor before updating) has been studied by Contreras and Tapia (1993), Burke and Weigmann (1997), and Kaufman (1999) in the context of trust-region methods.

### 8.4.2 Global Convergence

#### 8.4.2.1 Convergence to First-Order Critical Points

Because AM.3b and AM.4d are considerably weaker than AM.3 and AM.4, respectively, we need to review our convergence from the beginning, adapting our proofs as necessary. Of course, we need not revise the properties related to the decrease of the model (in Section 6.3), because, while our modified framework allows for more general models than before, it does not alter the magnitude of the reduction that can be obtained *once the model is defined*. We may therefore start our review with Section 6.4.

We first reexamine the size of the error between the true objective function and its model at a new iterate  $x_k + s_k$ , in the case where we assume AM.3b or AM.3c and AM.4d.

**Theorem 8.4.2** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3b, and AM.4d hold. Then

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\text{egg}} \kappa_{\text{une}} \|g_k\| \Delta_k + \kappa_{\text{ubh}} \varphi_k \Delta_k^2, \quad (8.4.11)$$

where

$$\kappa_{\text{ubh}} \stackrel{\text{def}}{=} \frac{1}{2}(\kappa_{\text{ufh}} + 1)\kappa_{\text{une}}^2 > 1.$$

**Proof.** We refer the reader to the proof of Theorem 6.4.1 (p. 133) for comparison. Using AF.1 and AM.1, we may apply the mean value theorem on the objective function and the model, and deduce that

$$f(x_k + s_k) = f(x_k) + \langle s_k, \nabla_x f(x_k) \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle \quad (8.4.12)$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ . Similarly, we have that

$$m_k(x_k + s_k) = m_k(x_k) + \langle s_k, g_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle \quad (8.4.13)$$

for some  $\zeta_k$  in the segment  $[x_k, x_k + s_k]$ .

Subtracting (8.4.13) from (8.4.12) and taking absolute values yields that

$$\begin{aligned} |f(x_k + s_k) - m_k(x_k + s_k)| &= |\langle s_k, \nabla_x f(x_k) - g_k \rangle \\ &\quad + \frac{1}{2} \langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle - \langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\ &\leq |\langle s_k, \nabla_x f(x_k) - g_k \rangle| \\ &\quad + \frac{1}{2} |\langle s_k, \nabla_{xx} f(\xi_k) s_k \rangle| + \frac{1}{2} |\langle s_k, \nabla_{xx} m_k(\zeta_k) s_k \rangle| \\ &\leq \|s_k\| \|v_k\| + \frac{1}{2} (\kappa_{\text{ufh}} + \varphi_k) \|s_k\|^2 \\ &\leq \kappa_{\text{egg}} \nu_k^{\text{s}} \|g_k\| \Delta_k + \frac{1}{2} (\kappa_{\text{ufh}} + \varphi_k) [\nu_k^{\text{s}}]^2 \Delta_k^2 \\ &\leq \kappa_{\text{egg}} \kappa_{\text{une}} \|g_k\| \Delta_k + \frac{1}{2} (\kappa_{\text{ufh}} + 1) \kappa_{\text{une}}^2 \varphi_k \Delta_k^2, \end{aligned}$$

where we successively used AM.2, AM.3b, (8.4.1), the triangle and Cauchy–Schwarz inequalities, AF.3, AM.4d, the fact that  $x_k + s_k \in \mathcal{B}_k$  implies that  $\|s_k\| \leq \nu_k^{\text{s}} \Delta_k$ , the bound  $\nu_k^{\text{s}} \leq \kappa_{\text{une}}$  (because of AN.1), and the fact that  $\varphi_k \geq 1$  by definition. Hence inequality (8.4.11) holds. Note that  $\kappa_{\text{ubh}} > 1$  because  $\kappa_{\text{ufh}} > 1$  and  $\kappa_{\text{une}} \geq 1$ .  $\square$

At variance with Theorem 6.4.1, the bounds on the error between the model and the objective function now explicitly include the model curvature  $\varphi_k$ . Moreover, the bound obtained also includes a first-order term in  $\Delta_k$ . As before, we now need to show that an iteration must be successful if the current iterate is not first-order critical and the trust-region radius is small enough.

**Theorem 8.4.3** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3b, AM.4d, and AA.1 hold. Suppose furthermore that  $g_k \neq 0$ . Then, if

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\|}{\kappa_{\text{ubh}} \varphi_k} \left[ 1 - \eta_2 - \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} \right], \quad (8.4.14)$$

we have that iteration  $k$  is successful and

$$\Delta_{k+1} \geq \Delta_k. \quad (8.4.15)$$

**Proof.** We refer the reader to the proof of Theorem 6.4.2 (p. 134), for comparison. The condition of  $\kappa_{\text{egg}}$  in AM.3b and the inequalities  $\kappa_{\text{mdc}} < 1$  and  $\kappa_{\text{ubh}} > 1$  imply that

$$\frac{\kappa_{\text{mdc}}}{\kappa_{\text{ubh}}} \left[ 1 - \eta_2 - \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} \right] \leq 1. \quad (8.4.16)$$

Thus condition (8.4.14) implies that

$$\Delta_k \leq \frac{\|g_k\|}{\varphi_k},$$

and AA.1 immediately gives that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\varphi_k}, \Delta_k \right] = \kappa_{\text{mdc}} \|g_k\| \Delta_k. \quad (8.4.17)$$

We may now apply Theorem 8.4.2 and deduce from (8.4.17), (8.4.11), (8.4.16), and (8.4.14) that

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k + s_k) - m_k(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \right| \\ &\leq \left| \frac{\kappa_{\text{egg}} \kappa_{\text{une}} \|g_k\| \Delta_k}{\kappa_{\text{mdc}} \|g_k\| \Delta_k} \right| + \left| \frac{\kappa_{\text{ubh}} \varphi_k \Delta_k^2}{\kappa_{\text{mdc}} \|g_k\| \Delta_k} \right| \\ &\leq \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} + \frac{\kappa_{\text{ubh}} \varphi_k \Delta_k}{\kappa_{\text{mdc}} \|g_k\|} \\ &\leq 1 - \eta_2. \end{aligned}$$

Therefore,  $\rho_k \geq \eta_2 \geq \eta_1$  and the iteration is very successful. Furthermore, the mechanism of the trust-region radius update (6.1.5) (p. 116) ensures that (8.4.15) holds.  $\square$

We conclude from this result that the radius cannot become too small relative to the curvature of the model so long as the algorithm stays away from first-order critical points. As before, this ensures that progress of the algorithm is always possible (except at first-order critical points).

**Theorem 8.4.4** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3b, AM.4d, and AA.1 hold. Suppose furthermore that there exists a constant  $\kappa_{\text{lbg}} > 0$  such that  $\|g_k\| \geq \kappa_{\text{lbg}}$  for all  $k$ . Then there is a constant  $\kappa_{\text{lbd}} > 0$  such that

$$\Delta_k \geq \frac{\kappa_{\text{lbd}}}{\varphi_k} \quad (8.4.18)$$

for all  $k$ .

**Proof.** Assume that iteration  $k$  is the first such that

$$\Delta_{k+1} \leq \frac{\gamma_1 \kappa_{\text{mdc}} \kappa_{\text{lbg}}}{\kappa_{\text{ubh}} \varphi_k} \left[ 1 - \eta_2 - \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} \right]. \quad (8.4.19)$$

Then we have from the mechanism of Algorithm BTR (see (6.1.5), p. 116) that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \kappa_{\text{lbg}}}{\kappa_{\text{ubh}} \varphi_k} \left[ 1 - \eta_2 - \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} \right],$$

and our assumption on  $\|g_k\|$  implies that (8.4.14) holds and thus that (8.4.15) is satisfied. This contradicts the fact that iteration  $k$  is the first such that (8.4.19) holds, and our initial assumption is therefore impossible. This yields the desired lower bound with

$$\kappa_{\text{lbd}} = \frac{\gamma_1 \kappa_{\text{mdc}} \kappa_{\text{lbg}}}{\kappa_{\text{ubh}}} \left[ 1 - \eta_2 - \frac{\kappa_{\text{egg}} \kappa_{\text{une}}}{\kappa_{\text{mdc}}} \right]. \quad \square$$

We now prove an analog of the weak convergence result Theorem 6.4.5 (p. 136) in the context of our more general framework for choosing the model.

**Theorem 8.4.5** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3b, AM.4d, AA.1, and AA.4 hold. Then one has that

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (8.4.20)$$

**Proof.** Assume, for the purpose of deriving a contradiction, that, for all  $k$ ,

$$\|g_k\| \geq \epsilon \quad (8.4.21)$$

for some  $\epsilon > 0$  and consider a successful iteration with index  $k$ . The fact that  $k \in \mathcal{S}$ , the sufficient-decrease requirement (6.3.27) (p. 131), and the bounds (8.4.10) and (8.4.21) then give that

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\varphi_k}, \Delta_k \right].$$

If we now sum over all successful iterations from 1 to  $k$ , we deduce that

$$f(x_1) - f(x_{k+1}) = \sum_{\substack{j=1 \\ j \in \mathcal{S}}}^k [f(x_j) - f(x_{j+1})] \geq \kappa_{\text{mdc}} \epsilon \eta_1 \sum_{\substack{j=1 \\ j \in \mathcal{S}}}^k \min \left[ \frac{\epsilon}{\varphi_j}, \Delta_j \right].$$

But AF.2 implies that the left-hand side of this inequality is bounded above and thus, using (8.4.18), that the sum

$$\min[\epsilon, \kappa_{\text{lbd}}] \sum_{\substack{j=1 \\ j \in \mathcal{S}}}^k \frac{1}{\varphi_j} = \sum_{\substack{j=1 \\ j \in \mathcal{S}}}^k \min \left[ \frac{\epsilon}{\varphi_j}, \frac{\kappa_{\text{lbd}}}{\varphi_j} \right] \leq \sum_{\substack{j=1 \\ j \in \mathcal{S}}}^k \min \left[ \frac{\epsilon}{\varphi_j}, \Delta_j \right]$$

converges to a finite limit. Now let  $p$  be an integer such that

$$\gamma_5 \gamma_2^{p-1} < 1, \quad (8.4.22)$$

which is possible because  $\gamma_2 < 1$ , and define

$$\sigma_k = |\mathcal{S} \cap \{1, \dots, k\}|,$$

the number of successful iterations up to iteration  $k$ . Then define

$$\mathcal{S}_1 = \{k \geq 1 \mid k \leq p\sigma_k\} \text{ and } \mathcal{S}_2 = \{k \geq 1 \mid k > p\sigma_k\},$$

where the indices in both sets are ordered by increasing value. We now want to show that both sums

$$\sum_{k \in \mathcal{S}_1} \frac{1}{\varphi_k} \text{ and } \sum_{k \in \mathcal{S}_2} \frac{1}{\varphi_k} \quad (8.4.23)$$

are finite. Consider the first. If it has only finitely many terms, its convergence is obvious. Otherwise, that is, if  $\mathcal{S}_1$  has infinitely many elements, we construct a further subsequence,  $\mathcal{S}_3$ , say, consisting of the set of indices  $k \in \mathcal{S}$  ( $k \geq 1$ ), in ascending order, with each index repeated  $p$  times.

Figures 8.4.1 and 8.4.2 illustrate the definitions of these sequences by an example. We observe that these definitions imply that the  $j$ th element of  $\mathcal{S}_3$  is not greater than the  $j$ th element of  $\mathcal{S}_1$ . This gives that

$$\sum_{k \in \mathcal{S}_1} \frac{1}{\varphi_k} \leq \sum_{k \in \mathcal{S}_3} \frac{1}{\varphi_k} = p \sum_{k \in \mathcal{S}} \frac{1}{\varphi_k} < \infty,$$

because of the nondecreasing nature of the sequence  $\{\varphi_k\}$  and the convergence of the last sum. We now turn our attention to the second sum in (8.4.23). Observe that, for  $k \in \mathcal{S}_2$ ,

$$\frac{\kappa_{\text{lbd}}}{\varphi_k} \leq \Delta_k \leq \gamma_5^{\sigma_k} \gamma_2^{k-\sigma_k} \Delta_0 \leq \gamma_5^{k/p} \gamma_2^{k-k/p} \Delta_0 = \left( \gamma_5 \gamma_2^{p-1} \right)^{k/p} \Delta_0,$$

where we have used (8.4.18), AA.4, and the definition of  $\mathcal{S}_2$ . This yields that

$$\sum_{k \in \mathcal{S}_2} \frac{1}{\varphi_k} \leq \frac{\Delta_0}{\kappa_{\text{lbd}}} \sum_{k \in \mathcal{S}_2} \left( \gamma_5 \gamma_2^{p-1} \right)^{k/p} < \infty,$$

and the second sum is convergent because of (8.4.22). Hence the sum

$$\sum_{k=1}^{\infty} \frac{1}{\varphi_k} = \sum_{k \in \mathcal{S}_1} \frac{1}{\varphi_k} + \sum_{k \in \mathcal{S}_2} \frac{1}{\varphi_k}$$

is finite, which contradicts AM.4d. As a consequence, we deduce that (8.4.21) is impossible and (8.4.20) follows.  $\square$

This theorem tells us that we can find a subsequence such that the corresponding iterates are asymptotically first-order critical for the model, and thus, as a consequence of Lemma 8.4.1, for the true objective function.

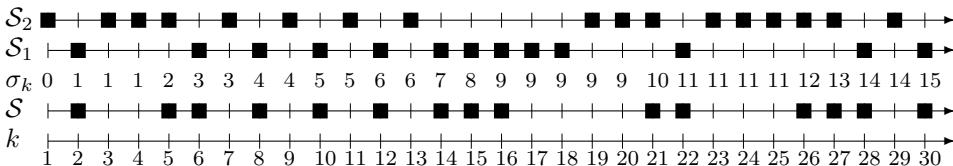


Figure 8.4.1: An example for the sequences  $\mathcal{S}$ ,  $\{\sigma_k\}$ ,  $\mathcal{S}_1$ , and  $\mathcal{S}_2$  (with  $p = 2$ ).

$\mathcal{S}_3$	2	2	5	5	6	6	8	8	10	10	12	12	14	...		
$\mathcal{S}_1$	2	6	8	10	12	14	15	16	17	18	22	28	30	...		
$\mathcal{S}$	2	5	6	8	10	12	14	15	16	21	22	26	27	28	30	...

Figure 8.4.2: The indices in the sequences  $\mathcal{S}$ ,  $\mathcal{S}_1$ , and  $\mathcal{S}_3$  corresponding to the example of Figure 8.4.1.

We next consider the case where there are only finitely many successful iterations.

**Theorem 8.4.6** Suppose that AF.1–AF.3, AM.1, AM.2, AM.3b, AM.4d, AA.1, and AA.4 hold. Suppose furthermore that there are only finitely many successful iterations. Then  $x_k = x_*$  for all sufficiently large  $k$  and  $x_*$  is first-order critical.

**Proof.** The mechanism of the algorithm ensures that  $x_* = x_{k_0+1} = x_{k_0+j}$  for all  $j > 0$ , where  $k_0$  is the index of the last successful iterate. Let us now denote by  $\mathcal{C}$  the index set of a subsequence of iterates such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{C}}} g_k = 0,$$

which is possible because of Theorem 8.4.5. We then deduce from Lemma 8.4.1, with  $\{k_i\} = \mathcal{C}$ , that

$$\|\nabla_x f(x_*)\| = \lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{C}}} \|\nabla_x f(x_k)\| = 0,$$

and thus  $x_*$  must be first-order critical.  $\square$

After completing the usual first step in the convergence analysis of a trust-region algorithm, we now wish to prove that, when there are infinitely many successful iterations, *all* limit points of the sequence of iterates are first-order critical. Unfortunately, this seems to require an additional assumption on the growth of the Hessian of the model, which appears to be somewhat unnatural in practical algorithms. We nevertheless consider it because it throws some light on the potential difficulties for algorithms using unbounded sequences of model Hessians. Furthermore, it is trivially satisfied when the Hessians are bounded, which is a practical setting even if the gradients are only approximate.

Thus we introduce the following additional assumption.

**AM.4f**

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{S}}} \varphi_k [f(x_k) - f(x_{k+1})] = 0.$$

Notice that AM.4f is in the same spirit as AM.4d combined with AA.4 in that it imposes a restriction on the relative rates of growth for  $\varphi_k$  and  $\Delta_k$ . Unfortunately, it is impossible to enforce in a practical algorithm, unless the model Hessians are not known to remain bounded, since it requires the algorithm to limit the size of the model Hessian at iteration  $k$  based on the knowledge of the as-yet-undetermined  $(k + 1)$ st iterate. Despite the theoretical nature of this assumption, we use it to derive the desired stronger convergence result.

**Theorem 8.4.7** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3b, AM.4d, AM.4f, AA.1, and AA.4 hold. Suppose moreover that there are infinitely many successful iterations. Then

$$\lim_{k \rightarrow \infty} \|g_k\| = \lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

**Proof.** We may suppose, for the purpose of obtaining a contradiction, that there exists a subsequence of successful iterates indexed by  $\{t_i\}$  such that

$$\|\nabla_x f(x_{t_i})\| \geq \epsilon_0 \quad (8.4.24)$$

for some  $\epsilon_0 > 0$  and for all  $i$ . Then, because of Lemma 8.4.1, we obtain that

$$\|g_{t_i}\| \geq 2\epsilon > 0$$

for some  $\epsilon > 0$  and for all  $i$  sufficiently large. Without loss of generality, we may assume that

$$(2 + \kappa_{\text{egg}})\epsilon \leq \frac{1}{2}\epsilon_0. \quad (8.4.25)$$

Theorem 8.4.5 then guarantees the existence for each  $t_i$  of a first successful iteration  $\ell(t_i) > t_i$  such that  $\|g_{\ell(t_i)}\| < \epsilon$ . Denoting  $\ell_i \stackrel{\text{def}}{=} \ell(t_i)$ , we thus deduce that there exists another subsequence of  $\mathcal{S}$  indexed by  $\{\ell_i\}$  such that

$$\|g_k\| \geq \epsilon \text{ for } t_i \leq k < \ell_i \text{ and } \|g_{\ell_i}\| < \epsilon. \quad (8.4.26)$$

We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\text{def}}{=} \{k \in \mathcal{S} \mid t_i \leq k < \ell_i\},$$

where  $t_i$  and  $\ell_i$  belong to the two subsequences defined above.<sup>132</sup> Using the sufficient decrease condition (6.3.27) (p. 131), the fact that  $\mathcal{K} \subseteq \mathcal{S}$ , and (8.4.26), we obtain that, for  $k \in \mathcal{K}$ ,

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\text{mdc}}\epsilon\eta_1 \min\left[\frac{\epsilon}{\varphi_k}, \Delta_k\right]. \quad (8.4.27)$$

Multiplying both sides of this inequality by  $\varphi_k$ , we obtain that

$$\varphi_k[f(x_k) - f(x_{k+1})] \geq \kappa_{\text{mdc}}\epsilon\eta_1 \min[\epsilon, \varphi_k \Delta_k].$$

As a consequence of AM.4f, we then obtain that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \varphi_k \Delta_k = 0,$$

---

<sup>132</sup>Figure 6.4.1 (p. 138) illustrates the definition of the subsequences of this theorem. These definitions are identical to that used in Theorem 6.4.6 (p. 137).

and the second term therefore dominates in the minimum of (8.4.27). We thus obtain from this latter inequality that, for  $k \in \mathcal{K}$  sufficiently large,

$$\Delta_k \leq \frac{1}{\kappa_{\text{mdc}} \epsilon \eta_1} [f(x_k) - f(x_{k+1})].$$

We then deduce from this bound and AN.1 that, for  $i$  sufficiently large,

$$\|x_{t_i} - x_{\ell_i}\| \leq \sum_{\substack{j=t_i \\ j \in \mathcal{K}}}^{\ell_i-1} \|x_j - x_{j+1}\| \leq \sum_{\substack{j=r_i \\ j \in \mathcal{K}}}^{\ell_i-1} \nu_j^s \Delta_j \leq \frac{\kappa_{\text{une}}}{\kappa_{\text{mdc}} \epsilon \eta_1} [f(x_{t_i}) - f(x_{\ell_i})]. \quad (8.4.28)$$

Now using AF.2 and the monotonicity of the sequence  $\{f(x_k)\}$ , we see that the right-hand side of inequality (8.4.28) must converge to zero, and we therefore obtain that

$$\lim_{i \rightarrow \infty} \|x_{t_i} - x_{\ell_i}\| = 0. \quad (8.4.29)$$

On the other hand, we have that

$$\|\nabla_x f(x_{t_i})\| \leq \|\nabla_x f(x_{t_i}) - \nabla_x f(x_{\ell_i})\| + \|v_{\ell_i}\| + \|g_{\ell_i}\|.$$

The first term of the right-hand side tends to zero because of the continuity of the gradient (AF.1) and (8.4.29). It is thus bounded by  $\epsilon$  for  $i$  sufficiently large. The third term is bounded by  $\epsilon$  using (8.4.26). This latter relation also implies, together with AM.3b, that

$$\|v_{\ell_i}\| \leq \kappa_{\text{egg}} \epsilon$$

and therefore, combining this inequality with (8.4.26) and (8.4.24), that

$$\|\nabla_x f(x_{t_i})\| \leq (2 + \kappa_{\text{egg}}) \epsilon \leq \frac{1}{2} \epsilon_0$$

for  $i$  large enough. Here we have used (8.4.25) to obtain the second inequality. But this contradicts (8.4.24), and this latter inequality must therefore be false, which proves that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

The proof of the theorem is then completed by invoking Lemma 8.4.1.  $\square$

This proves global convergence to first-order critical points.

#### 8.4.2.2 Convergence to Second-Order Critical Points

We may also consider extending the convergence results of Sections 6.5 and 6.6 to account for approximate model derivatives. This requires the following stronger condition on the error  $v_k$ .

**AM.3c** There exists a constant  $\kappa_{\text{egg}}$  such that

$$\|v_k\| \leq \kappa_{\text{egg}} \min(\|g_k\|, \|\nabla_x f(x_k)\|)$$

for all  $k$  and

$$0 \leq \kappa_{\text{egg}} < \min \left[ \frac{1}{2}, \frac{\kappa_{\text{mdc}}(1 - \eta_2)}{\kappa_{\text{une}}} \right]. \quad (8.4.30)$$

We now start by proving a version of Theorem 6.5.2 (p. 141) showing that the complete sequence of iterates converges to a single first-order critical point when the model Hessians are asymptotically convex along a converging subsequence.

**Theorem 8.4.8** Suppose that AF.1–AF.3, AN.1, AM.1, AM.2, AM.3c, AM.4d, AM.4f, AA.1, and AA.4 hold and that  $\{x_{k_i}\}$  is a subsequence of the iterates generated by Algorithm BTR converging to the first-order critical point  $x_*$ . Suppose furthermore that there is a constant  $\kappa_{\text{smh}} > 0$  such that

$$\liminf_{i \rightarrow \infty} \min_{x \in \mathcal{B}_{k_i}} \lambda_1[\nabla_{xx} m_{k_i}(x)] \geq \kappa_{\text{smh}}. \quad (8.4.31)$$

Suppose finally that  $\nabla_{xx} f(x_*)$  is nonsingular. Then the complete sequence of iterates  $\{x_k\}$  converges to  $x_*$ .

**Proof.** We refer the reader to the proof of Theorem 6.5.2 (p. 141), for comparison. As in the earlier theorem, we note that the criticality of  $x_*$  is ensured by Theorem 8.4.7 and that the conclusion is obvious if there are only finitely many successful iterations. Otherwise, we assume that the subsequence  $\{k_i\}$  is infinite and consists of successful iterations only, yielding

$$x_{k_i+1} = x_{k_i} + s_{k_i} \quad (8.4.32)$$

for all  $i$ . In order to prove the convergence of the complete sequence, we choose a  $\delta > 0$ , whose value is small enough to ensure that

$$\|\nabla_{xx} f(x) - \nabla_{xx} f(x_*)\| \leq \frac{1}{4} \min[1, \sigma, \kappa_{\text{smh}}] \stackrel{\text{def}}{=} \delta_0 \quad (8.4.33)$$

for all  $x$  such that  $\|x - x_*\| \leq \delta$ , where  $\sigma > 0$  is the smallest singular value of  $\nabla_{xx} f(x_*)$ . This is possible because of the nonsingularity of  $\nabla_{xx} f(x_*)$  and the continuity of the Hessian of the objective function (AF.1). We may also choose  $i_1$  large enough to ensure that

$$\min_{x \in \mathcal{B}_{k_i}} \lambda_1[\nabla_{xx} m_{k_i}(x)] \geq \frac{\kappa_{\text{smh}}}{2}, \quad (8.4.34)$$

and

$$\|x_{k_i} - x_*\| \leq \frac{\kappa_{\text{smh}} \delta}{4\delta_0 + \kappa_{\text{smh}}} \stackrel{\text{def}}{=} \delta_1 \quad (8.4.35)$$

for all  $i \geq i_1$ , and also that

$$\|g_k\| \leq \delta_0 \delta_1 < \delta \quad (8.4.36)$$

for all  $k \geq k_{i_1}$ . Inequality (8.4.34) is possible because of (8.4.31) and (8.4.35) by the assumption that  $\{x_{k_i}\}$  converges to  $x_*$ . Inequality (8.4.36) follows because of Theorem 8.4.7 and the inequalities

$$\delta_0 < 1 \text{ and } \delta_1 < \delta,$$

which themselves result from the definitions of  $\delta_0$  and  $\delta_1$  in (8.4.33) and (8.4.35), respectively.

As in the earlier theorem again, we now deduce from Lemma 6.5.1 (p. 140) at iteration  $k_i$  with  $\epsilon = \frac{1}{2}\kappa_{\text{smh}}$  that

$$\|s_{k_i}\| \leq \frac{4}{\kappa_{\text{smh}}} \|g_{k_i}\|. \quad (8.4.37)$$

As a consequence, combining (8.4.32), (8.4.35), (8.4.36), and (8.4.37), we obtain that

$$\|x_{k_i+1} - x_*\| \leq \|x_{k_i} - x_*\| + \|s_{k_i}\| \leq \left(1 + \frac{4\delta_0}{\kappa_{\text{smh}}}\right) \delta_1 = \delta. \quad (8.4.38)$$

Assume now that

$$\|x_{k_i+1} - x_*\| > \delta_1, \quad (8.4.39)$$

and observe that

$$\nabla_x f(x_{k_i+1}) = \nabla_x f(x_*) + G_*(x_{k_i+1} - x_*),$$

where

$$G_* = \int_0^1 \nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) dt. \quad (8.4.40)$$

Hence, using the triangle inequality, the definition of  $\sigma$ , (8.4.38), and (8.4.39),

$$\begin{aligned} \|\nabla_x f(x_{k_i+1})\| &= \|\nabla_{xx} f(x_*)(x_{k_i+1} - x_*) + (G_* - \nabla_{xx} f(x_*))(x_{k_i+1} - x_*)\| \\ &\geq \|\nabla_{xx} f(x_*)(x_{k_i+1} - x_*)\| - \|G_* - \nabla_{xx} f(x_*)\| \|(x_{k_i+1} - x_*)\| \\ &\geq \sigma \|x_{k_i+1} - x_*\| - \|G_* - \nabla_{xx} f(x_*)\| \|(x_{k_i+1} - x_*)\| \\ &> \sigma \delta_1 - \|G_* - \nabla_{xx} f(x_*)\| \delta. \end{aligned} \quad (8.4.41)$$

But (8.4.40) and (8.4.33) then give that

$$\begin{aligned} \|G_* - \nabla_{xx} f(x_*)\| &= \left\| \int_0^1 [\nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) - \nabla_{xx} f(x_*)] dt \right\| \\ &\leq \max_{t \in [0,1]} \|\nabla_{xx} f(x_{k_i+1} + t(x_* - x_{k_i+1})) - \nabla_{xx} f(x_*)\| \\ &\leq \delta_0. \end{aligned} \quad (8.4.42)$$

Therefore, combining (8.4.1), AM.3c, (8.4.41), (8.4.42), the definition of  $\delta_1$  in

(8.4.35), the definition of  $\delta_0$  in (8.4.33), and the bound (8.4.30), we deduce that

$$\begin{aligned}
\|g_{k_i+1}\| &\geq \|\nabla_x f(x_{k_i+1})\| - \|v_{k_i+1}\| \\
&\geq (1 - \kappa_{\text{egg}}) \|\nabla_x f(x_{k_i+1})\| \\
&> (1 - \kappa_{\text{egg}}) [\sigma\delta_1 - \delta_0\delta] \\
&\geq (1 - \kappa_{\text{egg}}) \frac{\delta_1\sigma}{4} \left( 4 - \frac{4\delta_0 + \kappa_{\text{smh}}}{\kappa_{\text{smh}}} \right) \\
&\geq (1 - \kappa_{\text{egg}}) \delta_0\delta_1 \frac{3\kappa_{\text{smh}} - 4\delta_0}{\kappa_{\text{smh}}} \\
&\geq 2(1 - \kappa_{\text{egg}})\delta_0\delta_1 \\
&\geq \delta_0\delta_1.
\end{aligned} \tag{8.4.43}$$

This is impossible because of (8.4.36), and therefore

$$\|x_{k_i+1} - x_*\| \leq \delta_1.$$

All the conditions that are satisfied at  $x_{k_i}$  thus remain satisfied at  $x_{k_i+1}$ , and the argument can be applied recursively to show that, for all  $j \geq 1$ ,

$$\|x_{k_i+j} - x_*\| \leq \delta_1 < \delta.$$

Since  $\delta$  is arbitrarily small, this proves the convergence of the complete sequence  $\{x_k\}$  to  $x_*$ .  $\square$

The reason for the factor of  $\frac{1}{2}$  in (8.4.30) appears in the last inequality of (8.4.43). Modifying the coefficient of  $\kappa_{\text{smh}}$  in (8.4.33) shows that any value strictly smaller than 0.6 would also be adequate.

It is natural to ask at this stage if pursuing our extensions of the convergence results beyond Theorem 8.4.8 is worthwhile, since such extensions would require AM.5 or some adaptation of it. This may appear to be in contradiction with our, thus far minimal, requirements on the model Hessians (see AM.4d and AM.4f) and the possibility of allowing for inexact gradients. However, two reasons motivate us to continue our investigation. The first is that AM.5 need only hold when the model gradients converge to zero, which is, in view of AM.3b, in the neighbourhood of first-order critical points. But we know from the same AM.3b that the model gradients become asymptotically exact in these circumstances. This opens the possibility that Hessian approximations based on gradient values (such as the finite-difference methods we shall consider in the next section) may in turn be sufficiently accurate. The second reason is that quasi-Newton matrices may themselves converge to the Hessian of the objective function as first-order critical points are approached.<sup>133</sup> We therefore see

---

<sup>133</sup>This was shown to be true for the safeguarded SR1 update by Conn, Gould, and Toint (1991a) and Byrd, Khalbafan, and Schnabel (1996) under additional assumptions. See also Khalbafan, Byrd, and Schnabel (1999), where a different update is used at unsuccessful iterations.

that AM.5 is far from being unreasonable in our context. Moreover, AM.6 is also acceptable in the framework of quasi-Newton algorithms, because the model Hessian does not change within the trust region, which obviously ensures that it is Lipschitz continuous. Finally, as AA.2 is an assumption on the step  $s_k$  for a given model, it does not depend on the exact nature of this model. We thus believe that investigating the theoretical consequences of assumptions AM.5, AM.6, and AA.2 in the context of approximate models is relevant to practical computations.

Fortunately, this exploration can be carried out with little additional effort, for a detailed analysis of the proofs of Lemmas 6.5.3, 6.5.4, and 6.6.6 and Theorems 6.5.5, 6.6.5, 6.6.7, and 6.6.8 shows that these results remain valid without any modification, given the conclusion of Theorem 8.4.7.

## Notes and References for Subsection 8.4.2

Various conditions have been given in the literature under which the convergence of methods using models with approximate gradients may be investigated. The first such condition was suggested by Moré (1983), where the condition

$$\lim_{k \rightarrow \infty} \|v_k\| = 0 \text{ whenever } \{x_k\} \text{ is convergent}$$

was required to obtain global convergence to first-order critical points. However, this condition did not imply any particular mechanism specifying the desired accuracy on the gradient calculation at any given iteration, which may be considered a practical drawback. In an attempt to improve on this situation, the conditions

$$\|v_k\| \leq \kappa_{\text{egd}} \Delta_k \text{ or } \|v_k\| \leq \kappa_{\text{egd}} \Delta_k \|\nabla_x f(x_k)\|$$

were introduced by Toint (1988) and subsequently used by Conn et al. (1993) and Conn, Gould, Sartenaer, and Toint (1996b). They have also been used, seemingly independently, by Felgenhauer (1997) in a similar analysis. The condition that we have chosen to use in this exposition, AM.3b, was proposed and analysed by Carter (1991). Interestingly, Carter also required that the step  $s_k$  be asymptotically aligned with the negative gradient when the trust-region radius converges to zero. This assumption is not severe, for it is satisfied for a number of existing algorithms for computing  $s_k$  in the unconstrained case. But it is not without drawbacks if one wishes to apply the same idea in the context of problems with bound constraints (see Conn et al., 1993 for a discussion of this point). Note that our presentation does not require this additional hypothesis.

The use of possibly unbounded sequences of Hessian approximation has been mostly motivated by the success, for both small and large problems, of algorithms based on quasi-Newton updates. Among these techniques, the famous BFGS update was independently proposed by Broyden (1970), Fletcher (1970c), Goldfarb (1970), and Shanno (1970). The SR1 update was first suggested in Davidon (1968) and partly revived by Conn, Gould, and Toint (1991a). Limited-memory quasi-Newton methods were proposed in a series of papers by Perry (1976), Shanno (1978), Buckley (1978a, 1978b), Nocedal (1980), and Liu and Nocedal (1989) (see also Nocedal, 1986, for a specialized trust-region algorithm for this case, as well as Nash and Nocedal, 1991). Partitioned quasi-Newton techniques were introduced by Griewank and Toint (1982a, 1982b) and were implemented in Toint (1983a, 1983b) and Conn, Gould, and Toint

(1992b). See also Zhu, Nazareth, and Wolkowicz (1999) for a restricted form of quasi-Newton updating and its relationship to trust-region methods, and Chen, Deng, and Zhang (1995) for the case of unary functions.<sup>134</sup> A slightly stronger form of AM.4d was first introduced by Powell (1975) for obtaining the result that at least one limit point is first-order critical. This form was subsequently weakened in Toint (1981a, 1988) to obtain the formulation used here. The additional condition

$$\lim_{k \rightarrow \infty} \varphi_k[f(x_k) - f(x_{k+1})] = 0$$

(which is equivalent to AM.4f because  $x_k = x_{k+1}$ , and thus  $f(x_k) = f(x_{k+1})$ , at unsuccessful iterations) was proposed by Toint (1988) to obtain that all limit points share this desirable first-order criticality property. Conn, Vicente, and Visweswarah (1999) also suggested the condition

$$f(x_k) - f(x_{k+1}) \geq \kappa_{\text{amg}} \|g_k\| \|x_k - x_{k+1}\|,$$

but the gradients have to be exact to prove the desired convergence result. Furthermore, the circumstances in which this condition might be expected to hold appear to be more restrictive than for AM.4f. The condition certainly holds when the step  $s_k$  is aligned with the steepest-descent direction and the objective function is convex.

The introduction of AA.4 and AM.4f raises some additional questions. The first is whether any of these assumptions is actually vital to derive our global convergence results. One may also wonder if Theorem 8.4.5 can be proved under AM.4f but without AA.4. To our knowledge, these questions are at present unresolved.

We finally mention that an alternative way of handling quasi-Newton methods within trust-region methods is not to modify the convergence theory by introducing AM.4d/AM.4f, but rather to modify the quasi-Newton updating formula to ensure that these matrices remain bounded, in which case the theory of Chapter 6 directly applies. This is the approach proposed by Carter (1987).

### 8.4.3 Finite-Difference Approximations to Derivatives

A common way to compute approximate first and second derivatives is to use what are called *finite-difference approximations*.

#### 8.4.3.1 Finite-Difference Gradients

Let us examine approximating the gradient first. Formally, we may define the finite-difference approximation of the gradient  $\nabla_x f(x)$  componentwise by the expression

$$[\bar{g}(x, h)]_i = \frac{f(x + [h]_i e_i) - f(x)}{\|[h]_i\|} \quad (i = 1, \dots, n), \quad (8.4.44)$$

where  $h \in \mathbb{R}^n$  is a vector of “difference stepsizes”. This is known as a *forward finite-difference approximation*. Elementary analysis shows that

$$\nabla_x f(x) = \lim_{\|h\| \rightarrow 0} \bar{g}(x, h),$$

---

<sup>134</sup>That is, a function of the form  $f(x) = h(\langle a, x \rangle)$ , where  $h : \mathbb{R} \rightarrow \mathbb{R}$  and  $a \in \mathbb{R}^n$ .

which indicates that  $\bar{g}(x, h)$  is likely a reasonable approximation of the gradient  $\nabla_x f(x)$  when  $\|h\|$  is sufficiently small. This approximation requires  $n$  additional function evaluations (at  $x + [h]_i e_i$ ) if we assume that  $f(x)$  is known. In fact, we recall from Theorem 3.1.6 (p. 29) and the definition of  $\kappa_{\text{ufh}}$  (p. 121) that

$$\|\nabla_x f(x) - \bar{g}(x, h)\| \leq \frac{1}{2} \kappa_{\text{ufh}} \|h\|.$$

At the  $k$ th iterate, this is nothing but

$$\|v_k\| \leq \frac{1}{2} \kappa_{\text{ufh}} \|h_k\|$$

so long as we choose  $g_k = \bar{g}(x_k, h_k)$ . Enforcing AM.3b is therefore not difficult, at least in theory. Indeed, it is enough to require that

$$\|h_k\| \leq \frac{2\kappa_{\text{egg}}}{\kappa_{\text{ufh}}} \|g_k\|. \quad (8.4.45)$$

However, the value of  $\|g_k\|$  is only known after  $\bar{g}(x_k, h_k)$  has been evaluated. This means that, in theory again, we might have to use a simple iteration to determine the desired approximation, reducing  $\|h\|$  until (8.4.45) holds or  $\|g_k\|$  is zero.

One could also use the more accurate *central finite-difference approximation* given by

$$[\hat{g}(x, h)]_i = \frac{f(x + [h]_i e_i) - f(x - [h]_i e_i)}{2|[h]_i|} \quad (i = 1, \dots, n), \quad (8.4.46)$$

for which an error bound of the form

$$\|\nabla_x f(x) - \hat{g}(x, h)\| \leq \frac{1}{6} \kappa_{\text{ufn}} \|h\|^2$$

can be shown. However, (8.4.46) implies that  $2n$  additional function evaluations are necessary to obtain an approximate gradient, instead of the  $n$  for (8.4.44). This extra expense is often considered to be excessive, at least in regions where  $\|g_k\|$  is not small.

In practice, the situation is different because rounding errors in the calculation of expression (8.4.44) prevent  $\bar{g}(x, h)$  from being as accurate as one might wish. Indeed, these errors may even dominate the result completely for very small values of  $|[h]_i|$ . The problem arises because  $f(x + [h]_i e_i)$  is then very close to  $f(x)$  (or even identical, for  $[h]_i$  smaller than the relative machine precision), which causes a loss in the number of significant digits of the difference  $f(x + [h]_i e_i) - f(x)$ , itself magnified by the division by  $|[h]_i|$ . One thus aims at some compromise between large rounding errors (small  $h$ ) and large approximation errors (large  $h$ ): extensive numerical experience indicates that choosing  $|[h]_i|$  to be on the order of the square root of the machine precision typically yields a value of  $\bar{g}(x, h)$  of the same order of accuracy if the evaluation of  $f(x)$  can be achieved with an accuracy of the order of the machine precision. For the forward difference scheme (8.4.44), a popular choice for  $[h]_i$  is

$$[h]_i = \text{sign}(x_i) \sqrt{\epsilon_{\text{M}}} \max[|x_i|, S_{ii}], \quad (8.4.47)$$

where  $S$  is a diagonal scaling matrix as in Section 6.7. This choice has a number of advantages. The first is that the sign of  $[h]_i$  is chosen such that the addition  $x + [h]_i e_i$

involves quantities of the same sign, which is better for retaining as many significant digits as possible. In the same vein, the second advantage is that the size of  $[h]_i$  is now proportional to  $x_i$  itself, which prevents rounding errors due to a vast difference in the orders of magnitude of these two quantities. Finally, the inclusion of the “typical” value of  $x_i$  provides a safeguard when this component is too close to zero. A nice technical trick is to recompute the stepsize from

$$[h]_i = (x_i + [h]_i e_i) - x_i$$

after  $x_i + [h]_i e_i$  has been calculated. This improves the accuracy slightly. It may also be advantageous to abandon (8.4.44) when  $\|g_k\|$  is small, and use (8.4.46) instead. In this case, the recommended stepsizes are

$$[h]_i = \text{sign}(x_i) \epsilon_M^{1/3} \max[|x_i|, S_{ii}]. \quad (8.4.48)$$

This strategy is used by several optimization packages.

If we now return to our considerations of convergence theory, we immediately notice that the limitations of finite-precision calculations will, in practice, prevent (8.4.45) from holding for values of  $\|g_k\|$  smaller than the square root of machine precision  $\epsilon_M$ . Hence this level of accuracy on the gradient appears to be the limit of what can be reached by Algorithm BTR (p. 116) when finite-difference gradients are used, unless function values are computed in higher precision. Furthermore, there is no real advantage to using a larger value of  $\|h\|$  when  $\|g_k\|$  itself is larger: one still needs to evaluate the objective function at  $n$  additional points to compute  $\bar{g}(x, h)$ , and choosing a small  $h$  (as specified by (8.4.47)) ensures that no reduction of  $\|h\|$  will be necessary to enforce (8.4.45). As a consequence, the choice (8.4.47) is normally made at every iteration of Algorithm BTR, even if larger values could in principle be allowed by condition (8.4.45).

### 8.4.3.2 Finite-Difference Hessians

Finite-difference approximations are also possible for second derivative matrices. They come as two kinds: differences in function values and differences in gradients, when these are analytically available. In the first case, the  $(i, j)$ th element of the Hessian is approximated by the formula

$$[\bar{H}]_{ij} = \frac{f(x + [h]_i e_i + [h]_j e_j) - f(x + [h]_i e_i) - f(x + [h]_j e_j) + f(x)}{|[h]_i [h]_j|} \quad (8.4.49)$$

for  $i, j = 1, \dots, n$ . Taking symmetry into account, this requires  $\frac{1}{2}n(n+1)$  additional function evaluations to that of  $f(x)$ . Obviously, this cost is relatively high,<sup>135</sup> and this strategy is therefore only of practical use when the cost of evaluating the objective function is low or when a massively parallel computer is being used. The recommended value of the difference stepsizes are now given by

$$[h]_i = \text{sign}(x_i) \epsilon_M^{1/4} \max[|x_i|, S_{ii}]. \quad (8.4.50)$$

---

<sup>135</sup>This is why more elaborate difference schemes are usually judged too expensive for this calculation.

Figure 8.4.3 illustrates the approximation errors as a function of  $\|h\|$  for the gradient using forward and central differences (left picture) and for the Hessian using (8.4.49) (right picture) in IEEE double-precision arithmetic. The example taken is the calculation of the first and second derivatives of the one-dimensional sine function at 1. The horizontal axis shows the logarithm<sup>136</sup> of  $\|h\|$ , while the logarithm of the error is plotted on the vertical axis. In the left picture, three vertical lines also indicate the values of  $\epsilon_M$ ,  $\epsilon_M^{1/2}$ , and  $\epsilon_M^{1/3}$  (from left to right). In the right picture, the values of  $\epsilon_M$  and  $\epsilon_M^{1/4}$  are shown in the same manner. It is interesting to note that the values of  $\|h\|$  recommended above produce an error that is close to the minimum possible. The higher accuracy obtained for central compared to forward differences is very apparent in the left picture. The role of rounding errors for small  $h$  is also clearly visible, as they cause the jagged error curves on the left sides of both pictures.

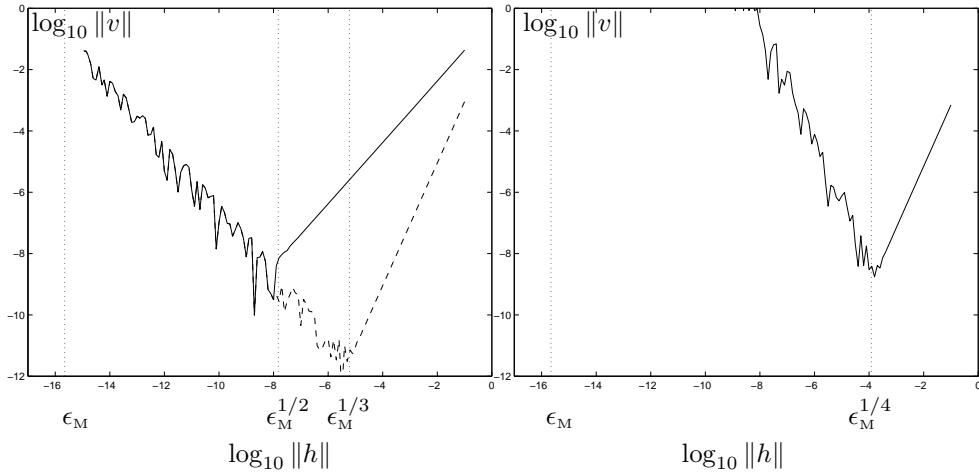


Figure 8.4.3: Left: the value of the logarithm of error on the gradient as a function of the logarithm of  $\|h\|$ , for the forward (solid) and central (dashed) difference schemes. Right: the value of the logarithm of the error on the Hessian as a function of the logarithm of  $\|h\|$  for (8.4.49).

We may also obtain an approximation of the Hessian matrix  $\nabla_{xx}f(x)$  by taking differences in the gradients, when these are available. Then one would typically compute the  $(i, j)$ th entry of this approximation by

$$[\bar{H}^{(0)}]_{ij} = \frac{[g(x + [h]_i e_i)]_j - [g(x)]_j}{|[h]_i|} \quad (i, j = 1, \dots, n), \quad (8.4.51)$$

or

$$[\hat{H}^{(0)}]_{ij} = \frac{[g(x + [h]_i e_i)]_j - [g(x - [h]_i e_i)]_j}{2|[h]_i|} \quad (i, j = 1, \dots, n),$$

followed by the symmetrization

$$\bar{H} = \frac{1}{2}(\bar{H}^{(0)} + [\bar{H}^{(0)}]^T) \text{ or } \hat{H} = \frac{1}{2}(\hat{H}^{(0)} + [\hat{H}^{(0)}]^T).$$

<sup>136</sup>In base 10.

In this case, the recommended difference stepsizes are also given by (8.4.47) and the cost of approximating one Hessian matrix is  $n$  gradients, in addition to the calculation of  $g(x)$ . For reasons of cost similar to those discussed above for differences in function values, central differences in the gradient are typically not computed.

At variance with the first derivatives, none of our assumptions on approximate Hessians require that they be asymptotically exact, at least as far as convergence to first-order critical points is concerned. But this requirement (AM.5) reappears if we wish to consider convergence to second-order critical points, as one must guarantee that any negative curvature present in  $\nabla_{xx}f(x_k)$  is reflected in  $\nabla_{xx}m_k(x_k)$ . This again means that the difference stepsizes have to converge to zero, which we have seen to be unrealistic in finite-precision calculations. Some (small) negative curvature in the Hessian of the objective function may therefore remain undetected when it is approximated by finite differences.

All our considerations on finite differences have so far assumed that the objective function (in (8.4.44), (8.4.46), and (8.4.49)) or the gradient (in (8.4.51)) can be computed with a relative accuracy comparable to  $\epsilon_M$ . If additional noise affects these evaluations, such that the relative precision on  $f(x)$  or  $g(x)$  is  $\theta > \epsilon_M$ , then the values of  $\epsilon_M$  should be replaced by  $\theta$  in (8.4.47), (8.4.48), and (8.4.50).

We conclude this paragraph by some general comments on the use of finite-difference approximations to derivatives. The first is that their practical relevance may decrease considerably in the next few years, because of the wider availability of automatic differentiation packages. These very useful tools accept as input the code for computing  $f(x)$  and then provide, at reasonable cost, the values of the analytic derivatives at  $x$  at the same time as the value of  $f(x)$ . This is an important development in many applications and is currently offered as an option by some recent optimization codes. One expects the use of these techniques to become even more widespread, which may ultimately make finite-difference techniques—indeed, quasi-Newton methods in general—obsolete. The second is that it is often preferable to use quasi-Newton approximations to the Hessian of the objective function than to approximate it using differences. This latter procedure appears to be justifiable only if the cost of the differencing remains small, because those of evaluating the objective or its gradient are themselves small.

## Notes and References for Subsection 8.4.3

Finite-difference approximations to gradients and Hessians have been popular in optimization algorithms for a long time. Clearly, their use is not restricted to the framework of trust regions, but may be applied much more generally. One of the first proposals in this area can be found in Stewart (1967), where the use of finite-difference approximations of the gradient is related to that of a quasi-Newton scheme for the Hessian. The books of Dennis and Schnabel (1983, Sections 5.4 and 5.6) and Gill, Murray, and Wright (1981, Sections 4.6.1.3 and 8.6) and the paper of Gill et al. (1983) present an interesting discussion of the topic, including details of how such approximations can be included in practical algorithms. We also mention here the remarkably efficient methods proposed by Curtis, Powell, and Reid (1974), Coleman and Moré (1983), and Goldfarb and Toint (1984) for the estimation of sparse Jacobian matrices

by finite differences, and the corresponding algorithms suggested in Powell and Toint (1979) and in Coleman and Moré (1984) for the approximation of Hessians. The main feature of all these methods is a careful exploitation of sparsity in order to significantly reduce the number of differences that are needed for approximating the complete Jacobian or Hessian.

Automatic differentiation is also an old and important idea, although it has come of age only relatively recently. An introduction to the subject may be found in the survey papers by Griewank (1989, 1994) or in the volume edited by Griewank and Corliss (1991).

## 8.5 Composite Problems and Models

In our introductory discussion of AM.1–AM.4 (p. 122), we mentioned the possibility that the objective function could be of the “composite” form

$$f(x) = f_0(x, u(x)), \quad (8.5.1)$$

where  $f_0(x, u)$  is a function from  $\mathbb{R}^n \times \mathbb{R}^p$  into  $\mathbb{R}$  which is relatively simple to compute, but where the vector-valued function  $u(x)$  from  $\mathbb{R}^n$  to  $\mathbb{R}^p$  is complicated and costly. In this case, we considered a “composite” model of the form

$$m_k(x) = f_0(x, m_k^u(x)), \quad (8.5.2)$$

where  $m_k^u(x)$  models  $u(x)$  in the neighbourhood of  $x_k$ . Maybe the simplest occurrence of this situation is when the objective is the sum of an “easy” part and a “complicated” part, that is,

$$f(x) = f_0(x) + u(x),$$

where  $f_0$  is “easy” and  $u$  is “complicated”. We would then have that (8.5.2) has the form

$$m_k(x) = f_0(x) + m_k^u(x).$$

For the general form (8.5.1)–(8.5.2), we now examine the reformulation of our assumptions AF.1–AF.3, AM.1–AM.6, and AA.1–AA.3 in this context. We consider first the assumptions we made to ensure convergence of Algorithm BTR (p. 116) to first-order critical points.

### Theorem 8.5.1

- (i) Suppose that  $f_0(x, u)$  is twice-continuously differentiable with respect to  $x$  and  $u$  and that AF.1 holds with  $f(x)$  replaced by  $u(x)$ . Then AF.1 holds.
- (ii) Suppose that  $f_0(x, u)$  is bounded below for all  $x \in \mathbb{R}^n$  and all  $u \in \mathbb{R}^p$ . Then AF.2 holds.
- (iii) Suppose that the second derivatives of  $f_0(x, u)$  and  $u(x)$ , together with  $\nabla_x u(x)$  and  $\nabla_u f_0(x, u)$ , are bounded above by a constant independent of  $x$  and  $u$ . Then AF.3 holds.

- (iv) Suppose that AM.1 holds with  $m_k$  replaced by  $m_k^u$  and that  $f_0(x, u)$  is twice-continuously differentiable with respect to  $x$  and  $u$ . Then AM.1 holds.
- (v) Suppose that AM.2 and AM.3 hold with  $m_k$  replaced by  $m_k^u$ . Then AM.2 and AM.3 hold.
- (vi) Suppose that, for all  $k$ , the second derivatives of  $f_0(x, u)$  and  $m_k^u(x)$ , together with  $\nabla_x m_k^u(x)$  and  $\nabla_u f_0(x, u)$ , are bounded above in  $\mathcal{B}_k$  by a constant independent of  $x$  and  $u$ . Then AM.4 holds.

**Proof.** The assertions (i)–(iii) merely indicate that the assumptions on the problem have to be stated in terms of  $f_0(x, y)$  and  $u(x)$  instead of  $f(x)$ , as expected. They trivially follow from the definitions.

Observe now that the gradient of the objective function is given by

$$\nabla_x f(x) = \nabla_x f_0(x, u(x)) + \nabla_x u(x) \nabla_u f_0(x, u(x)),$$

and that of the model by

$$\nabla_x m_k(x) = \nabla_x f_0(x, m_k^u(x)) + \nabla_x m_k^u(x) \nabla_u f_0(x, m_k^u(x)). \quad (8.5.3)$$

As a consequence, if AM.2 and AM.3 hold for the model  $m_k^u$ , that is, if

$$u(x_k) = m_k^u(x_k) \text{ and } \nabla_x u(x_k) = \nabla_x m_k^u(x_k),$$

then we obtain that

$$f(x_k) = m_k(x_k) \text{ and } \nabla_x f(x_k) = \nabla_x m_k(x_k),$$

which is nothing but AM.2 and AM.3 for  $m_k$ , proving (v).

Assertion (iv) results from the identity

$$\begin{aligned} \nabla_{xx} f(x) &= \nabla_{xx} f_0(x, u(x)) + [\nabla_x u(x)]^T \nabla_{ux} f_0(x, u(x)) \\ &\quad + \nabla_{xu} f_0(x, u(x)) \nabla_x u(x) + \nabla_{xx} u(x) \nabla_u f_0(x, u(x)) \\ &\quad + [\nabla_x u(x)]^T \nabla_{uu} f_0(x, u(x)) \nabla_x u(x). \end{aligned} \quad (8.5.4)$$

Finally, since

$$\begin{aligned} \nabla_{xx} m_k(x) &= \nabla_{xx} f_0(x, m_k^u(x)) + [\nabla_x m_k^u(x)]^T \nabla_{ux} f_0(x, m_k^u(x)) \\ &\quad + \nabla_{xu} f_0(x, m_k^u(x)) \nabla_x m_k^u(x) + \nabla_{xx} m_k^u(x) \nabla_u f_0(x, m_k^u(x)) \\ &\quad + [\nabla_x m_k^u(x)]^T \nabla_{uu} f_0(x, m_k^u(x)) \nabla_x m_k^u(x), \end{aligned} \quad (8.5.5)$$

one immediately obtains (vi).  $\square$

Notice that  $\nabla_{xx}u(x)$  and  $\nabla_{xx}m_k^u(x)$  are three-dimensional tensors. Observe also the requirements that  $\nabla_u f_0(x, u)$ ,  $\nabla_x u(x)$ , and  $\nabla_x m_k^u(x)$  all be uniformly bounded, although no specific assumption on the gradient norms is made to enforce convergence to first-order critical points in Chapter 6. Of course, these requirements are only needed if  $\nabla_{ux}f_0(x, m_k^u(x))$  or  $\nabla_{uu}f_0(x, u(x))$  are not identically zero, as we can see from equations (8.5.4) and (8.5.5). Finally, note that more complicated assumptions can be made to ensure AF.3 and AM.4 by insisting that growth in specific factors in the terms of (8.5.4) or (8.5.5) be compensated by a corresponding decrease in others, but this will not be discussed here.

The reformulation of the assumptions that are sufficient to ensure convergence to second-order critical points follows a similar pattern and requires the same additional conditions on uniform boundedness of gradients. The formal proof, however, requires a slightly more technical development.

**Theorem 8.5.2**

- (i) Suppose that

$$\lim_{k \rightarrow \infty} \|\nabla_{xx}u(x) - \nabla_{xx}m_k^u(x)\| = 0$$

whenever

$$\lim_{k \rightarrow \infty} \|\nabla_x f_0(x, m_k^u(x_k)) + \nabla_x m_k^u(x_k) \nabla_u f_0(x, u(x_k))\| = 0.$$

Then AM.5 holds.

- (ii) Suppose that, for all  $k$ , the second derivatives of  $f_0(x, u)$  and  $m_k^u(x)$  are Lipschitz continuous with respect to  $x$  and  $u$ . Suppose furthermore that, for all  $k$ ,  $\nabla_x m_k^u(x)$  and  $\nabla_u f_0(x, u)$  are bounded above in  $\mathcal{B}_k$  by a constant independent of  $x$  and  $u$ . Then AM.6 holds.

**Proof.** Assertion (i) is a direct consequence of (8.5.4) and (8.5.5) and the definition of  $g_k = \nabla_x m_k(x_k)$  via (8.5.3).

From (8.5.5), we obtain that

$$\begin{aligned} \|\nabla_{xx}m_k(x) - \nabla_{xx}m_k(z)\| &\leq \|\nabla_{xx}f_0(x, m_k^u(x)) - \nabla_{xx}f_0(z, m_k^u(z))\| \\ &\quad + \|[\nabla_x m_k^u(x)]^T \nabla_{ux}f_0(x, m_k^u(x)) - [\nabla_x m_k^u(z)]^T \nabla_{ux}f_0(z, m_k^u(z))\| \\ &\quad + \|\nabla_{xu}f_0(x, m_k^u(x)) \nabla_x m_k^u(x) - \nabla_{xu}f_0(z, m_k^u(z)) \nabla_x m_k^u(z)\| \\ &\quad + \|\nabla_{xx}m_k^u(x) \nabla_u f_0(x, m_k^u(x)) - \nabla_{xx}m_k^u(z) \nabla_u f_0(z, m_k^u(z))\| \\ &\quad + \|[\nabla_x m_k^u(x)]^T \nabla_{uu}f_0(x, m_k^u(x)) \nabla_x m_k^u(x) \\ &\quad \quad - [\nabla_x m_k^u(z)]^T \nabla_{uu}f_0(z, m_k^u(z)) \nabla_x m_k^u(z)\|. \end{aligned}$$

If  $\|\nabla_x m_k^u(x)\| \leq \kappa_{\text{umg}}$  and  $\|\nabla_u f_0(x, m_k^u(x))\| \leq \kappa_{\text{ufu}}$  for all  $x$ , this yields that

$$\begin{aligned}
\|\nabla_{xx}m_k(x) - \nabla_{xx}m_k(z)\| &\leq \|\nabla_{xx}f_0(x, m_k^u(x)) - \nabla_{xx}f_0(z, m_k^u(z))\| \\
&\quad + 2\kappa_{\text{umg}}\|\nabla_{xu}f_0(x, m_k^u(x)) - \nabla_{xu}f_0(z, m_k^u(z))\| \\
&\quad + \kappa_{\text{ufu}}\|\nabla_{xx}m_k^u(x) - \nabla_{xx}m_k^u(z)\| \\
&\quad + \kappa_{\text{umg}}^2\|\nabla_{uu}f_0(x, m_k^u(x)) - \nabla_{uu}f_0(z, m_k^u(z))\| \\
&\leq \|\nabla_{xx}f_0(x, m_k^u(x)) - \nabla_{xx}f_0(z, m_k^u(x))\| \\
&\quad + \|\nabla_{xx}f_0(z, m_k^u(x)) - \nabla_{xx}f_0(z, m_k^u(z))\| \\
&\quad + 2\kappa_{\text{umg}}\|\nabla_{xu}f_0(x, m_k^u(x)) - \nabla_{xu}f_0(z, m_k^u(x))\| \\
&\quad + 2\kappa_{\text{umg}}\|\nabla_{xu}f_0(z, m_k^u(x)) - \nabla_{xu}f_0(z, m_k^u(z))\| \\
&\quad + \kappa_{\text{ufu}}\|\nabla_{xx}m_k^u(x) - \nabla_{xx}m_k^u(z)\| \\
&\quad + \kappa_{\text{umg}}^2\|\nabla_{uu}f_0(x, m_k^u(x)) - \nabla_{uu}f_0(z, m_k^u(x))\| \\
&\quad + \kappa_{\text{umg}}^2\|\nabla_{uu}f_0(z, m_k^u(x)) - \nabla_{uu}f_0(z, m_k^u(z))\|.
\end{aligned}$$

Let  $\kappa_{\text{lfx}}$ ,  $\kappa_{\text{lfu}}$ , and  $\kappa_{\text{lfm}}$  be the Lipschitz constants of  $\nabla_{xx}f_0(x, u)$  with respect to  $x$ , of  $\nabla_{xx}f_0(x, u)$  with respect to  $u$ , and of  $\nabla_{xx}m_k^u(x)$ , respectively. We may then deduce from the last inequality that

$$\begin{aligned}
\|\nabla_{xx}m_k(x) - \nabla_{xx}m_k(z)\| &\leq \kappa_{\text{lfx}}\|x - z\| + \kappa_{\text{lfu}}\|m_k^u(x) - m_k^u(z)\| \\
&\quad + 2\kappa_{\text{umg}}\kappa_{\text{lfx}}\|x - z\| + 2\kappa_{\text{umg}}\kappa_{\text{lfu}}\|m_k^u(x) - m_k^u(z)\| \\
&\quad + \kappa_{\text{ufu}}\kappa_{\text{lfm}}\|x - z\| + \kappa_{\text{umg}}^2\kappa_{\text{lfx}}\|x - z\| \\
&\quad + \kappa_{\text{umg}}^2\kappa_{\text{lfu}}\|m_k^u(x) - m_k^u(z)\| \\
&= [\kappa_{\text{lfx}}(1 + \kappa_{\text{umg}})^2 + \kappa_{\text{ufu}}\kappa_{\text{lfm}}]\|x - z\| \\
&\quad + \kappa_{\text{lfu}}(1 + \kappa_{\text{umg}})^2\|m_k^u(x) - m_k^u(z)\|.
\end{aligned}$$

Now applying the mean value theorem, we obtain that

$$\|m_k^u(x) - m_k^u(z)\| \leq \|\nabla_x m_k^u(\zeta)\| \|x - z\| \leq \kappa_{\text{umg}}\|x - z\|,$$

where  $\zeta \in [x, z]$ . Hence we finally conclude that

$$\begin{aligned}
\|\nabla_{xx}m_k(x) - \nabla_{xx}m_k(z)\| &\leq [\kappa_{\text{lfx}}(1 + \kappa_{\text{umg}})^2 + \kappa_{\text{ufu}}\kappa_{\text{lfm}}]\|x - z\| \\
&\quad + \kappa_{\text{lfu}}(1 + \kappa_{\text{umg}})^2\kappa_{\text{umg}}\|x - z\| \\
&= [(\kappa_{\text{lfx}} + \kappa_{\text{lfu}}\kappa_{\text{umg}})(1 + \kappa_{\text{umg}})^2 + \kappa_{\text{lfx}}\kappa_{\text{ufu}}]\|x - z\|,
\end{aligned}$$

which is AM.6 with

$$\kappa_{\text{lch}} = (\kappa_{\text{lfx}} + \kappa_{\text{lfu}}\kappa_{\text{umg}})(1 + \kappa_{\text{umg}})^2 + \kappa_{\text{lfx}}\kappa_{\text{ufu}}.$$

Thus (ii) follows.  $\square$

We conclude this analysis by observing that AA.1, AA.2, and AA.3 do not need to be reformulated in the framework considered here because they do not directly depend on the form of the objective function or of the model. Theorems 8.5.1 and 8.5.2 together thus give a set of assumptions on the composite objective function and model (8.5.4) and (8.5.5) that ensure the convergence of Algorithm BTR (p. 116) to second-order critical points.

Finally, it is of interest to note that the composite unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f_0(x, u(x))$$

may also be seen as a constrained problem of the form

$$\min_{x \in \mathbb{R}^n} f_0(x, u) \text{ subject to the constraint that } u = u(x).$$

Trust-region methods for handling this alternative formulation will be considered in Chapters 14 and 15.

# Chapter 9

---

## Conditional Models

---

### 9.1 Motivation and Formal Description

Having examined the impact of using approximate derivatives, we now turn to a different kind of approximation of the objective function by a model. More precisely, we examine the situation where the model  $m_k$  can only be considered as a suitable approximation of the objective function when some additional condition is satisfied. In this chapter, we will consider two cases of interest, in which additional conditions like these occur quite naturally. The first is when the model is determined by a multivariate interpolation scheme, using objective function values at several distinct interpolation points. This arises when we wish to use a trust-region optimization method that does not require derivatives of the objective function, a surprisingly common situation in practice. In this case, the model derived from interpolation is invalid unless some geometrical conditions on the interpolation points hold. For instance, six points on a straight line do not determine a full quadratic model in two dimensions. We thus have to require that the interpolation points somehow “fill the space” in a sense we shall formally discuss in Section 9.4. A second situation where the model is not unconditionally valid is when one tries to build a global model of the objective function, in which case the global nature of the model may be incompatible with the local behaviour of the objective function. Again, we will explain this in more detail in Section 9.5. For now, we shall merely presume that there are practical situations of interest where AM.2 or AM.3 (or both) may not be valid at every iteration, and therefore where Theorem 6.4.1 (p. 133) does not automatically hold.

The reader may wonder about the implications of the possibility that the model may not be valid at every iteration. As we have indicated above, the central point is that the error between the objective function and its model may be arbitrarily large within the trust region, that is, that Theorem 6.4.1 fails and thus that the ratio  $\rho_k$  of predicted versus achieved reduction may not be a reliable indicator of the need to reduce the trust-region radius. This has two serious consequences. Firstly, reducing the radius at an iteration where the model is invalid may fail to improve the validity

of the model. Secondly, it may have the undesirable effect of shortening subsequent steps more than necessary and, ultimately, of convergence of the sequence of iterates to a point that is not first-order critical. This is of course not what we want, and we must therefore revise the convergence theory we developed in Chapter 6.

Another observation is also in order. Clearly, the concept of the trust region itself is questionable if the model and the objective function do not ultimately agree as the radii shrink, because the convergence of an algorithm based on this idea then appears to be unlikely. It is therefore natural to require that, if the model is invalid at a given iteration, this situation should not last forever. In other words, it is crucial to ensure the validity of the model after a certain (finite) number of *model improvement steps*.

These two assumptions, namely, that the model is only valid if a specific condition is satisfied and that this condition must eventually hold after a finite number of improvement steps, form the basis of the developments presented in the remainder of this section. We call this kind of model a *conditional* model.

We now formalize our framework as follows. We say that the model  $m_k$  is *valid* in

$$\mathcal{Q}_k(\delta) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid \|x - x_k\| \leq \delta\} \quad (9.1.1)$$

whenever

$$|f(x) - m_k(x)| \leq \kappa_{\text{cnd}} \delta^2 \quad (9.1.2)$$

for all  $x \in \mathcal{Q}_k(\delta)$  and for some constant<sup>137</sup>  $\kappa_{\text{cnd}} > 0$  independent of  $x$ . Notice that this is a general form of the model error implied by Theorem 6.4.1 (p. 133).

Since  $\delta$  is, for now, arbitrary, we consider that the model  $m_k$  might be evaluated outside  $\mathcal{B}_k$ , and therefore extend our assumptions on the model to make these evaluations well defined.

**AM.1b** For all  $k$ , the model  $m_k$  is twice differentiable on  $\mathbb{R}^n$ .

**AM.4g** The Hessian of the model remains bounded on  $\mathbb{R}^n$ ; that is,

$$\|\nabla_{xx} m_k(x)\| \leq \kappa_{\text{umh}} - 1 \text{ for all } x \in \mathbb{R}^n$$

for all  $k$ .

This definition and new assumptions are motivated by the following result.

**Theorem 9.1.1** Suppose AF.1 and AF.3 hold together with AM.1b and AM.4g, and that  $m_k$  is valid in  $\mathcal{Q}_k(\delta)$  defined in (9.1.1). Then

$$\|\nabla_x f(x_k) - g_k\| \leq \kappa_{\text{aeg}} \delta \quad (9.1.3)$$

for some constant<sup>138</sup>  $\kappa_{\text{aeg}} > 0$  independent of  $k$ .

<sup>137</sup>“cnd” stands for “conditional error”.

<sup>138</sup>“aeg” stands for “absolute error on the gradient”.

**Proof.** If  $g_k = \nabla_x f(x_k)$ , then (9.1.3) trivially follows. Otherwise, since both the model  $m_k$  and the objective function are twice-continuously differentiable by AF.1 and AM.1b, Taylor's theorem gives, for every  $h$  satisfying the bound  $\|h\| \leq \delta$ , that

$$f(x_k + h) = f(x_k) + \langle \nabla_x f(x_k), h \rangle + \frac{1}{2} \langle h, \nabla_{xx} f(\xi_k) h \rangle$$

and

$$m_k(x_k + h) = m_k(x_k) + \langle g_k, h \rangle + \frac{1}{2} \langle h, \nabla_{xx} m_k(\zeta_k) h \rangle,$$

where  $\xi_k$  and  $\zeta_k$  belong to the segment  $[x_k, x_k + h]$ . Taking the difference of these two equations, we deduce that

$$\begin{aligned} \langle \nabla_x f(x_k) - g_k, h \rangle &= f(x_k + h) - m_k(x_k + h) + m_k(x_k) - f(x_k) \\ &\quad - \frac{1}{2} \langle h, \nabla_{xx} f(\xi_k) h \rangle + \frac{1}{2} \langle h, \nabla_{xx} m_k(\zeta_k) h \rangle, \end{aligned}$$

and therefore, using (9.1.2), AF.3, AM.4g, and the Cauchy–Schwarz inequality, that

$$\begin{aligned} |\langle \nabla_x f(x_k) - g_k, h \rangle| &\leq 2\kappa_{\text{cnd}}\delta^2 + \frac{1}{2}|\langle h, \nabla_{xx} f(\xi_k) h \rangle| + \frac{1}{2}|\langle h, \nabla_{xx} m_k(\zeta_k) h \rangle| \\ &\leq (2\kappa_{\text{cnd}} + \kappa_{\text{ubh}})\delta^2. \end{aligned}$$

Choosing

$$h = \delta \frac{\nabla_x f(x_k) - g_k}{\|\nabla_x f(x_k) - g_k\|},$$

we obtain that

$$\|\nabla_x f(x_k) - g_k\| \leq (2\kappa_{\text{cnd}} + \kappa_{\text{ubh}})\delta.$$

This is (9.1.3) with  $\kappa_{\text{aeg}} = 2\kappa_{\text{cnd}} + \kappa_{\text{ubh}}$ . □

In order to use the notion of a valid model in our framework, we furthermore assume the following.

**AM.7** The validity of  $m_k$  in  $\mathcal{Q}_k(\delta)$  may be checked at each iteration and for any value of  $\delta > 0$ , if required.

**AM.8** The model  $m_k$  can be made valid in  $\mathcal{Q}_k(\delta)$  in a finite number of *model improvement steps* for any  $k$  and any  $\delta > 0$ .

Note that we do not specify the nature of the model improvement steps at this stage, but merely assume that they are possible. For instance, if we consider the context of derivative-free optimization mentioned above, an improvement step may consist of evaluating the objective function at a new point  $y \in \mathcal{Q}_k(\delta)$ . Details of the improvement steps will be discussed for each of the applications of our framework.

Notice that (9.1.2) applies to all points within the trust region, and thus in particular at  $x_k$  and  $x_k + s_k$ . We may thus relax AM.2 along with AM.3, allowing for a model value that does not exactly reproduce those of the true objective function.

We may now formally state a conditional version of Algorithm BTR (p. 116).

**Algorithm 9.1.1: Conditional trust-region algorithm**

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0 = \Delta_{\text{ref}}$  are given, as are the constants

$$\epsilon_c > 0, \quad \alpha \in (0, 1), \quad \text{and} \quad \mu > 0.$$

The constants  $\eta_0$ ,  $\eta_1$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) as well as  $0 < \eta_0 \leq \eta_1$ . Compute  $f(x_0)$ , define a model  $m_0$ , and set  $k = 0$ .

**Step 1: Final criticality test.** If  $\|g_k\| \leq \epsilon_c$ , test if  $m_k$  is valid in  $\mathcal{Q}_k(\delta_k)$  for some  $\delta_k \in (0, \mu\|g_k\|]$ . If this is not the case, perform as many improvement steps as necessary to ensure that the model is valid in  $\mathcal{Q}_k(\alpha\mu\|g_k\|)$  and return to the beginning of Step 1.

**Step 2: Step calculation.** Choose  $\|\cdot\|_k$  and compute a step  $s_k$  that “sufficiently reduces the model”  $m_k$  (in the sense of AA.1) and such that  $x_k + s_k \in \mathcal{B}_k$ .

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $\mathcal{X}_k = \{x_k + s_k\}$ ; otherwise define  $\mathcal{X}_k = \{x_k\}$ .

**Step 4: Model improvement.** If  $\rho_k < \eta_1$  and  $m_k$  is invalid in  $\mathcal{B}_k$ , perform a model improvement step, possibly enlarging  $\mathcal{X}_k$ , and define  $m_{k+1}$  to be the improved model.

**Step 5: Selection of the next iterate.** Determine  $\hat{x}_k$  such that

$$f(\hat{x}_k) = \min_{x \in \mathcal{X}_k} f(x). \quad (9.1.4)$$

Then, if

$$\hat{\rho}_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(\hat{x}_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq \eta_0, \quad (9.1.5)$$

set  $x_{k+1} = \hat{x}_k$  and define a new model  $m_{k+1}$ . Otherwise, set  $x_{k+1} = x_k$ .

**Step 6: Trust-region radius update.** If  $\hat{\rho}_k \geq \eta_0$  or if  $m_k$  is valid in  $\mathcal{B}_k$ , set  $\Delta_{\text{ref}} = \Delta_k$ . Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_{\text{ref}}, \infty) & \text{if } \rho_k \geq \eta_1, \\ [\gamma_1 \Delta_{\text{ref}}, \gamma_2 \Delta_{\text{ref}}] & \text{if } \rho_k < \eta_1. \end{cases} \quad (9.1.6)$$

Increment  $k$  by 1 and go to Step 1.

The statement of this algorithm calls for several comments.

- (1) The algorithm depends crucially on AM.7, since the validity of the model is checked in Steps 1 (within  $\mathcal{Q}_k(\delta_k)$ ) and 4 (within  $\mathcal{B}_k$ ). As a consequence, we will be forced to assume AM.7 in all results for Algorithm 9.1.1.
- (2) It is important to note that improvement steps in Step 4 may involve the computation of the objective function value  $f(\bar{x}_k)$  for some  $\bar{x}_k \in \mathcal{B}_k$ , aside from  $x_k$  and  $x_k + s_k$ . In this case, the set  $\mathcal{X}_k$  is updated by

$$\mathcal{X}_k \leftarrow \mathcal{X}_k \cup \{\bar{x}_k\}.$$

This is in fact the reason for Step 5: any  $f(\bar{x}_k)$  that is better than  $f(x_k + s_k)$  may then be taken into account, and the corresponding  $\bar{x}_k$  considered as a potential next iterate.

This feature is also the reason why we have assumed that the model  $m_k$  is defined in the whole of  $\mathbb{R}^n$  (in AM.1b), not uniquely in  $\mathcal{B}_k$ . Indeed, until Step 5, we do not know whether or not the model improvement will generate a new and better point  $\bar{x}_k$ , in which case the next trust region will be centered at this new point<sup>139</sup> instead of being centered at  $x_k$ . As a consequence of this modification, the value of  $\|\cdot\|_k$  may only be selected at Step 2, before the computation of the step  $s_k$ .

- (3) Further improvement in the model, beyond that formally mentioned in the algorithm, is also possible. For example, one could decide to perform a model improvement step if  $\rho_k$  is very small, which indicates a bad fit of the model to the objective function. Any further decrease in function values obtained in this manner is then taken into account by Step 5.
- (4) The implicit loop in Step 1 may be viewed as a model improvement inner iteration, with the aim of ensuring that  $g_k$ , the first-order information for the model, is not too different from the true first-order information for the objective function. It remains to see whether this inner iteration is well defined and finite. We examine this question in Lemma 9.1.2.

In this inner iteration, the set  $\mathcal{Q}_k(\delta_k)$  plays a role similar to that of the trust region  $\mathcal{B}_k$ . Note that the additional computational effort required to obtain a valid model may be adjusted by a suitable choice of  $\epsilon_c$ .

Other ways to ensure that the gradient of the model is correctly approximated when it becomes small may of course also be considered.

- (5) Notice the simplified nature of the trust-region update mechanism, and, in particular, the dependence of  $\Delta_{k+1}$  on  $\Delta_{\text{ref}}$  rather than  $\Delta_k$ . This is the formal consequence of the observation that the trust-region radius should not be reduced

---

<sup>139</sup>Strictly speaking, it would be sufficient to assume that the model is defined in a trust region of radius  $2\Delta_{k+1}$ , but we have chosen to ignore this marginal and often irrelevant improvement.

too much if the model has not been guaranteed to be valid in  $\mathcal{B}_k$ . The new mechanism also guarantees that the radius does not decrease when the model is valid and fits well, an essential ingredient of the convergence theory (see Theorem 6.4.2 [p. 134]).

We now revise our definition of a successful iteration and say that iteration  $k$  is *successful* if the algorithm is able to progress, that is, if (9.1.5) holds. If (9.1.5) fails, then the iteration is *unsuccessful*. As before, we denote the index set of all successful iterations by  $\mathcal{S}$ . In other words,

$$\mathcal{S} = \{k \mid \hat{\rho}_k \geq \eta_0\}.$$

We also define  $\mathcal{R}$  to be the index set of all iterations where the reference trust-region radius  $\Delta_{\text{ref}}$  is reset at the beginning of Step 6, that is,

$$\mathcal{R} = \{k \mid \hat{\rho}_k \geq \eta_0 \text{ or } m_k \text{ is valid in } \mathcal{B}_k\}. \quad (9.1.7)$$

We conclude this introductory section by stating three useful properties of Algorithm 9.1.1.

**Lemma 9.1.2** Suppose AM.7 and AM.8 hold. Then we have the following results.

- (i) If  $\rho_k \geq \eta_1$ , then  $\hat{\rho}_k \geq \eta_0$ , and thus iteration  $k$  is successful. Moreover,  $\mathcal{S} \subseteq \mathcal{R}$ .
- (ii) There can only be a finite number of iterations such that  $\rho_k < \eta_1$  before  $\Delta_{\text{ref}}$  is set to  $\Delta_k$ .
- (iii) If the loop within Step 1 is infinite, then

$$\nabla_x f(x_k) = 0 \text{ and } \lim_{i \rightarrow \infty} \|g_k^{(i)}\| = 0,$$

where  $g_k^{(i)}$  are the gradients at  $x_k$  of the successive models computed in this loop.

**Proof.** If  $\rho_k \geq \eta_1$  the mechanism of Step 3 implies that  $x_k + s_k$  is added to the set  $\mathcal{X}_k$ . As a consequence, we have that

$$f(\hat{x}_k) \leq f(x_k + s_k)$$

because of the definition of  $\hat{x}_k$  in (9.1.4). Thus

$$\hat{\rho}_k = \frac{f(x_k) - f(\hat{x}_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} = \rho_k \geq \eta_1 \geq \eta_0,$$

and the first part of conclusion (i) follows. The second part then immediately results from the definition of  $\mathcal{R}$  in (9.1.7). Property (ii) follows from the mechanism of Step 4, AM.8, and the mechanism of Step 6. We must be more careful for (iii) because of the interaction between  $g_k$  and  $\delta_k$ . Assume, for the purpose of obtaining a contradiction, that the loop is infinite. At the start, we know that  $m_k$  is not valid

or that the radius of the neighbourhood of  $x_k$  in which it is valid exceeds  $\mu\|g_k\|$ . We then define  $g_k^{(0)} = g_k$  and the model is improved until it is valid in  $\mathcal{Q}_k(\alpha\mu\|g_k^{(0)}\|)$  (in a finite number of improvement steps because of AM.8). If the gradient of the resulting model  $m_k^{(1)}$ ,  $g_k^{(1)} = \nabla_x m_k^{(1)}(x_k)$ , satisfies  $\|g_k^{(1)}\| \geq \alpha\|g_k^{(0)}\|$ , the procedure stops with

$$\delta_k = \alpha\mu\|g_k^{(0)}\| \leq \mu\|g_k^{(1)}\|.$$

Otherwise, that is, if  $\mu\|g_k^{(1)}\| < \alpha\|g_k^{(0)}\|$ , the model is improved until it is valid in  $\mathcal{Q}_k(\alpha\mu\|g_k^{(1)}\|) \subset \mathcal{Q}_k(\alpha^2\mu\|g_k^{(0)}\|)$ . Then again, either the procedure stops or the radius of  $\mathcal{Q}_k$  is again multiplied by  $\alpha$ , and so on. The only way for this procedure to be infinite (and to require an infinite number of improvement steps) is if

$$\|g_k^{(i)}\| < \alpha^i\|g_k^{(0)}\| = \alpha^i\|g_k\|$$

for all  $i \geq 0$ , where  $g_k^{(i)} = \nabla_x m_k^{(i)}(x_k)$ . This shows the second part of (iii). But the model is then valid in each of the  $\mathcal{Q}_k(\alpha^i\mu\|g_k\|)$ , and therefore Theorem 9.1.1 implies that

$$\|\nabla_x f(x_k) - g_k^{(i)}\| \leq \kappa_{\text{aeg}}\alpha^i\mu\|g_k\|$$

for each  $i \geq 0$  sufficiently large, and thus, using the triangle inequality, that for all  $i \geq 0$  sufficiently large,

$$\|\nabla_x f(x_k)\| \leq \|g_k^{(i)}\| + \|\nabla_x f(x_k) - g_k^{(i)}\| \leq (1 + \kappa_{\text{aeg}}\mu)\alpha^i\|g_k\|.$$

Since  $\alpha \in (0, 1)$ , this implies that  $\nabla_x f(x_k) = 0$ , as desired.  $\square$

The first of these results indicates that the iteration has to be successful if the model fits the objective function well enough, as the reader might have expected. It also says that the model must be valid within  $\mathcal{B}_k$  for every iteration for which the trust-region radius is reduced. The third property states that if an infinite loop occurs within Step 1, this must be because the current iterate is first-order critical, in which case convergence to such a point is trivial. Note that in this case the number of successful iterations is finite. The second result guarantees the overall coherence of the method, in that the trust-region radius will ultimately be reduced if improvements to the model are not enough to ensure progress.

## Notes and References for Section 9.1

The notion of conditional models appears to be new but can be traced to the convergence theory proposed in Conn, Scheinberg, and Toint (1997a) for unconstrained derivative-free optimization.

## 9.2 Convergence to First-Order Critical Points

We now reexamine the convergence theory developed for Algorithm BTR (p. 116) and adapt it to Algorithm 9.1.1. The main differences between these two methods are of

course the condition for model validity, but there is also the simplified trust-region radius updating scheme and the presence of Step 5.

We first observe that the definitions of a valid model and its associated error bound (9.1.2) are very close in spirit to Theorem 6.4.1 (p. 133), if one assumes that  $\delta = \Delta_k$  and that the model is valid in  $\mathcal{B}_k$ . The last restriction is, however, important, and we will have to verify that we use the error bound only at iterations at which this validity condition holds. However, at such iterations, the smaller the radius becomes, the better the model approximates the objective function, which intuitively should guarantee that minimizing the model within the trust region will also decrease the value of the objective function, as desired. We now show that this intuition is vindicated, in that an iteration must be successful if the current iterate is not a first-order critical point and the trust-region radius is small enough (see Theorems 6.4.2 [p. 134] and 6.4.3 [p. 135]).

**Theorem 9.2.1** Suppose that AF.1–AF.3, AN.1, AM.1b, AM.4g, AM.7–AM.8, and AA.1 hold. Suppose furthermore that there exists a constant  $\kappa_{\text{lbg}} > 0$  such that  $\|g_k\| \geq \kappa_{\text{lbg}}$  for all  $k$ . Then there is a constant  $\kappa_{\text{lbd}} > 0$  such that

$$\Delta_k > \kappa_{\text{lbd}}$$

for all  $k$ .

**Proof.** Suppose that  $k$  is the first  $k \in \mathcal{R}$  such that

$$\Delta_k \leq \gamma_1 \min \left[ 1, \frac{\kappa_{\text{mde}} \kappa_{\text{lbg}} (1 - \eta_1)}{\kappa_{\text{ubh}}^2 \max[\kappa_{\text{ubh}}, 2\kappa_{\text{cnd}}]} \right]. \quad (9.2.1)$$

The mechanism of Step 6 then implies that there must exist an iteration  $r$  immediately preceding iteration  $k$  in the subsequence  $\mathcal{R}$  such that

$$\Delta_k \in [\gamma_1 \Delta_r, \gamma_2 \Delta_r]. \quad (9.2.2)$$

But, since iteration  $r$  is obviously not successful, we have that  $m_r$  is valid in  $\mathcal{B}_r$ . Furthermore, (9.2.1) and (9.2.2) also imply, from (9.1.6), that

$$\Delta_r \leq \min \left[ 1, \frac{\kappa_{\text{mde}} \kappa_{\text{lbg}} (1 - \eta_1)}{\max[\kappa_{\text{ubh}}, 2\kappa_{\text{cnd}}]} \right]. \quad (9.2.3)$$

Observe now that the condition  $\eta_1 \in (0, 1)$  and the inequality  $\kappa_{\text{mde}} < 1$  imply that

$$\kappa_{\text{mde}} (1 - \eta_1) < 1. \quad (9.2.4)$$

Thus condition (9.2.3) and our assumption on  $g_k$  imply that

$$\Delta_r \leq \frac{\|g_r\|}{\kappa_{\text{ubh}}}.$$

As a consequence, AA.1 immediately gives that

$$m_r(x_r) - m_r(x_r + s_r) \geq \kappa_{\text{mdc}} \|g_r\| \min \left[ \frac{\|g_r\|}{\kappa_{\text{ubh}}}, \Delta_r \right] = \kappa_{\text{mdc}} \|g_r\| \Delta_r. \quad (9.2.5)$$

On the other hand, since  $m_r$  is valid in  $\mathcal{B}_r$ , we may deduce from (9.2.5), (9.2.4), (9.2.3), and AN.1 that

$$\begin{aligned} |\rho_r - 1| &= \left| \frac{f(x_r) - f(x_r + s_r) - m_r(x_r) + m_r(x_r + s_r)}{m_r(x_r) - m_r(x_r + s_r)} \right| \\ &\leq \left| \frac{f(x_r) - m_r(x_r)}{m_r(x_r) - m_r(x_r + s_r)} \right| + \left| \frac{f(x_r + s_r) - m_r(x_r + s_r)}{m_r(x_r) - m_r(x_r + s_r)} \right| \\ &\leq \frac{2\kappa_{\text{cnd}} [\nu_k^{\text{S}}]^2}{\kappa_{\text{mdc}} \|g_k\|} \Delta_r \\ &\leq \frac{2\kappa_{\text{cnd}} \kappa_{\text{une}}^2}{\kappa_{\text{mdc}} \kappa_{\text{lbg}}} \Delta_r \\ &\leq 1 - \eta_1, \end{aligned}$$

where we have used (9.2.3) to deduce the last inequality. Therefore,  $\rho_r \geq \eta_1$  and, by Lemma 9.1.2 (i),  $\hat{\rho}_r \geq \eta_0$ . Iteration  $r$  is thus successful, which in turn implies that  $k = r + 1$  and  $\Delta_k \geq \Delta_r$ , which is a contradiction. As a consequence, there is no  $k \in \mathcal{R}$  such that (9.2.1) holds and

$$\Delta_{\text{ref}} \geq \gamma_1 \min \left[ 1, \frac{\kappa_{\text{mdc}} \kappa_{\text{lbg}} (1 - \eta_1)}{\max[\kappa_{\text{ubh}}, 2\kappa_{\text{cnd}}]} \right]$$

for all iterations. The mechanism of Step 6 then ensures that the desired conclusion holds with

$$\kappa_{\text{lbd}} = \gamma_1^2 \min \left[ 1, \frac{\kappa_{\text{mdc}} \kappa_{\text{lbg}} (1 - \eta_1)}{\max[\kappa_{\text{ubh}}, 2\kappa_{\text{cnd}}]} \right]. \quad \square$$

We have thus obtained a result similar to Theorem 6.4.3 (p. 135). We now continue our development, as in Chapter 6, by examining the case where there are only finitely many iterations.

**Theorem 9.2.2** Suppose that AF.1–AF.3, AM.1b, AM.4g, AM.7–AM.8, and AA.1 hold. Suppose furthermore that there are only finitely many successful iterations. Then  $x_k = x_*$  for all  $k$  sufficiently large and  $\nabla_x f(x_*) = 0$ .

**Proof.** If an infinite loop occurs in Step 1, the result follows from Lemma 9.1.2 (iii). Otherwise, the mechanism of the algorithm ensures that  $x_* = x_{k_0+1} = x_{k_0+j}$  for all  $j > 0$ , where  $k_0$  is the index of the last successful iteration. Moreover, since all iterations are unsuccessful for sufficiently large  $k$ , we obtain from Lemma 9.1.2 (i) that  $\hat{\rho}_k < \eta_0$ . The mechanism of Step 4 then implies that the model is improved at every iteration  $k \geq k_0$  and thus, because of AM.8, that there exists an infinite

subsequence  $\{k_j\}$  of unsuccessful iterations such that  $m_{k_j}$  is valid in  $\mathcal{B}_{k_j}$ . Furthermore, the mechanism of Step 6 then ensures that  $\{\Delta_k\}$ , and thus  $\{\Delta_{k_j}\}$ , converge to zero. Assume now that  $\nabla_x f(x_*) \neq 0$ . Using Theorem 9.1.1, we have that, for  $k_j > k_0$ ,

$$\|\nabla_x f(x_*) - g_{k_j}\| \leq \kappa_{\text{aeg}} \nu_{k_j} \Delta_{k_j},$$

which, since  $\{\Delta_{k_j}\}$  converges to zero, then implies that  $\|g_{k_j}\| \geq \frac{1}{2} \|\nabla_x f(x_*)\| > 0$  for  $k_j \geq k_0$  sufficiently large. But Theorem 9.2.1 states that  $\{\Delta_{k_j}\}$  cannot converge to zero in this case, yielding a contradiction. Hence  $\nabla_x f(x_*) = 0$ , as desired.  $\square$

Having proved a convergence property for the case where  $\mathcal{S}$  is finite, we may now verify that the gradient of the model is not bounded away from zero when  $\mathcal{S}$  is infinite.

**Theorem 9.2.3** Suppose that AF.1–AF.3, AN.1, AM.1b, AM.4g, and AA.1 hold. Then one has that

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0.$$

**Proof.** The proof of this result is identical to that of Theorem 6.4.5 (p. 136), except that we may no longer assume that the gradient of the model and that of the objective function coincide, which means that the assumption (6.4.16) must be replaced by

$$\|g_k\| \geq \kappa_{\text{lbg}}$$

for some  $\kappa_{\text{lbg}} > 0$ .  $\square$

We next verify that the mechanism of Step 1 ensures that the same result holds for the gradient of the objective function.

**Lemma 9.2.4** Suppose that AF.1, AF.3, AM.1b, AM.4g, and AM.7 hold. Suppose furthermore that  $\{k_i\}$  is a subsequence such that

$$\lim_{i \rightarrow \infty} \|g_{k_i}\| = 0. \quad (9.2.6)$$

Then one has that

$$\lim_{i \rightarrow \infty} \|\nabla_x f(x_{k_i})\| = 0. \quad (9.2.7)$$

**Proof.** By (9.2.6),  $\|g_{k_j}\| \leq \epsilon_c$  for  $j$  sufficiently large, and the mechanism of Step 1 ensures that  $m_{k_i}$  is valid in  $\mathcal{Q}_{k_i}(\delta_{k_i})$  for  $\delta_{k_i} \leq \mu \|g_{k_i}\|$  for  $i$  sufficiently large. Theorem 9.1.1 then allows us to deduce that, for  $i$  sufficiently large,

$$\|\nabla_x f(x_{k_i}) - g_{k_i}\| \leq \kappa_{\text{aeg}} \delta_{k_i} \leq \kappa_{\text{aeg}} \mu \|g_{k_i}\|. \quad (9.2.8)$$

As a consequence, we have that, for  $i$  sufficiently large,

$$\|\nabla_x f(x_{k_i})\| \leq \|g_{k_i}\| + \|\nabla_x f(x_{k_i}) - g_{k_i}\| \leq (1 + \kappa_{\text{aeg}}\mu)\|g_{k_i}\|.$$

The limit (9.2.6) and this last bound then give (9.2.7).  $\square$

Notice the similarity between inequality (9.2.8) and AM.3b: we may thus interpret Step 1 of Algorithm 9.1.1 as a particular way of enforcing AM.3b in the context of conditional models. Theorem 9.2.3 and Lemma 9.2.4 immediately give the following global convergence result.

**Theorem 9.2.5** Suppose that AF.1–AF.3, AN.1, AM.1b, AM.4g, AM.7–AM.8, and AA.1 hold. Then there is at least a subsequence of successful iterates  $\{x_k\}$  whose limit is a first-order critical point, that is,

$$\liminf_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

As always, we now prove that *all* limit points of the sequence of iterates are first-order critical. The proof of this result is very similar to that of Theorem 6.4.6 (p. 137), but we supply the details again for the sake of clarity.

**Theorem 9.2.6** Suppose that AF.1–AF.3, AN.1, AM.1b, AM.4g, AM.7–AM.8, and AA.1 hold. Then every limit point  $x_*$  of the sequence  $\{x_k\}$  is first-order critical, that is,  $\nabla_x f(x_*) = 0$ .

**Proof.** Suppose, for the purpose of establishing a contradiction, that there is a subsequence of successful iterates, indexed by  $\{t_i\} \subseteq \mathcal{S}$ , such that

$$\|\nabla_x f(x_{t_i})\| \geq \epsilon_0 > 0 \quad (9.2.9)$$

for some  $\epsilon_0 > 0$  and for all  $i$ . Then, because of Lemma 9.2.4, we obtain that

$$\|g_{t_i}\| \geq 2\epsilon > 0$$

for some  $\epsilon > 0$  and for all  $i$  sufficiently large. Without loss of generality, we may assume that

$$(2 + \kappa_{\text{aeg}}\mu)\epsilon \leq \frac{1}{2}\epsilon_0. \quad (9.2.10)$$

Theorem 9.2.3 then ensures the existence for each  $t_i$  of a first successful iteration  $\ell(t_i) > t_i$  such that  $\|g_{\ell(t_i)}\| < \epsilon$ . Denoting  $\ell_i \stackrel{\text{def}}{=} \ell(t_i)$ , we thus obtain that there exists another subsequence of  $\mathcal{S}$  indexed by  $\{\ell_i\}$  such that

$$\|g_k\| \geq \epsilon \text{ for } t_i \leq k < \ell_i \text{ and } \|g_{\ell_i}\| < \epsilon \quad (9.2.11)$$

for sufficiently large  $i$ . We now restrict our attention to the subsequence of successful iterations whose indices are in the set

$$\mathcal{K} \stackrel{\text{def}}{=} \{k \in \mathcal{S} \mid t_i \leq k < \ell_i\},$$

where  $t_i$  and  $\ell_i$  belong to the two subsequences defined above. Using AA.1, the fact that  $\mathcal{K} \subseteq \mathcal{S}$  and (9.2.11), we deduce that, for  $k \in \mathcal{K}$ ,

$$f(x_k) - f(x_{k+1}) \geq \eta_0[m_k(x_k) - m_k(x_k + s_k)] \geq \kappa_{\text{mdc}}\epsilon\eta_0 \min\left[\frac{\epsilon}{\kappa_{\text{ubh}}}, \Delta_k\right]. \quad (9.2.12)$$

But the sequence  $\{f(x_k)\}$  is monotonically decreasing and bounded below because of AF.2. Hence it is convergent, and the left-hand side of (9.2.12) must tend to zero when  $k$  tends to infinity. This gives that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \Delta_k = 0.$$

As a consequence, the second term asymptotically dominates in the minimum of (9.2.12), and we obtain that, for  $k \in \mathcal{K}$  sufficiently large,

$$\Delta_k \leq \frac{1}{\kappa_{\text{mdc}}\epsilon\eta_0}[f(x_k) - f(x_{k+1})].$$

We then deduce from this bound and AN.1 that, for  $i$  sufficiently large,

$$\|x_{t_i} - x_{\ell_i}\| \leq \sum_{\substack{j=t_i \\ j \in \mathcal{K}}}^{\ell_i-1} \|x_j - x_{j+1}\| \leq \kappa_{\text{une}} \sum_{\substack{j=t_i \\ j \in \mathcal{K}}}^{\ell_i-1} \Delta_j \leq \frac{\kappa_{\text{une}}}{\kappa_{\text{mdc}}\epsilon\eta_0}[f(x_{t_i}) - f(x_{\ell_i})].$$

Using AF.2 and the monotonicity of the sequence  $\{f(x_k)\}$  again, we see that the right-hand side of this inequality must converge to zero, and we therefore obtain that

$$\lim_{i \rightarrow \infty} \|x_{t_i} - x_{\ell_i}\| = 0.$$

Now

$$\|\nabla_x f(x_{t_i})\| \leq \|\nabla_x f(x_{t_i}) - \nabla_x f(x_{\ell_i})\| + \|\nabla_x f(x_{\ell_i}) - g_{\ell_i}\| + \|g_{\ell_i}\|.$$

The first term of the right-hand side tends to zero because of the continuity of the gradient of  $f$  (see AF.1) and is thus bounded by  $\epsilon$  for  $i$  sufficiently large. The third term is bounded by  $\epsilon$  by (9.2.11). Using a reasoning similar to that of Lemma 9.2.4, we also deduce that the second term is bounded by  $\kappa_{\text{aeg}}\mu\epsilon$  for  $i$  sufficiently large. As a consequence, we obtain from these bounds and (9.2.10) that

$$\|\nabla_x f(x_{t_i})\| \leq (2 + \kappa_{\text{aeg}}\mu)\epsilon \leq \frac{1}{2}\epsilon_0$$

for  $i$  large enough, which contradicts (9.2.9). Hence our initial assumption must be false and the theorem follows.  $\square$

### 9.3 Convergence to Second-Order Critical Points

The technique of proof for Theorem 9.1.1 suggests that it may be that it is not only the gradient of the model and that of the objective function that are close to each other for small  $\delta$ , but that the same may be true of their Hessians. Of course, this requires that we strengthen the notion of model validity we introduced on p. 308.

We say that the model  $m_k$  is *second-order valid* in  $\mathcal{Q}_k(\delta)$  whenever it is valid and

$$|f(x) - m_k(x)| \leq \kappa_{\text{cnd}} \delta^3 \quad (9.3.1)$$

for all  $x \in \mathcal{Q}_k(\delta)$ .

Besides this stronger validity, we also need to strengthen the smoothness conditions of the objective function and model to being thrice-continuously differentiable with bounded second and third derivatives.

**AF.1c** The objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is thrice-continuously differentiable on  $\mathbb{R}^n$ .

**AF.4** The third derivative tensor of the objective function is bounded on  $\mathbb{R}^n$ ; that is,

$$\|\nabla_{xxx} f(x)\| \leq \kappa_{\text{uft}} \quad \text{for all } x \in \mathbb{R}^n,$$

for all  $k$ , where  $\kappa_{\text{uft}} \geq 1$  is a constant<sup>140</sup> independent of  $k$ .

**AM.1c** For all  $k$ , the model  $m_k$  is thrice differentiable on  $\mathbb{R}^n$ .

**AM.9** The third derivative tensor of the model remains bounded on  $\mathbb{R}^n$ ; that is,

$$\|\nabla_{xxx} m_k(x)\| \leq \kappa_{\text{umt}} \quad \text{for all } x \in \mathbb{R}^n$$

for all  $k$ , where  $\kappa_{\text{umt}} \geq 1$  is a constant<sup>141</sup> independent of  $k$ .

With these strengthened assumptions, we may now prove a bound on the distance between the Hessian of the objective function and that of the model.

**Theorem 9.3.1** Suppose AF.1c and AF.4 hold together with AM.1c and AM.9, and that  $m_k$  is second-order valid in  $\mathcal{Q}_k(\delta)$ . Then

$$\|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| \leq \kappa_{\text{aeh}} \delta \quad (9.3.2)$$

for some constant<sup>142</sup>  $\kappa_{\text{aeh}} > 0$  independent of  $k$ .

<sup>140</sup>“uft” stands for “upper bound on the objective function’s third derivative”.

<sup>141</sup>“umt” stands for “upper bound on the model’s third derivative”.

<sup>142</sup>“aeh” stands for “absolute error on the Hessian”.

**Proof.** The proof is similar in spirit to that of Theorem 9.1.1. If  $\nabla_{xx}m_k(x_k)$  is equal to  $\nabla_{xx}f(x_k)$  then (9.3.2) trivially follows. Otherwise, since both the model  $m_k$  and the objective function are thrice-continuously differentiable by AF.1c and AM.1c, Taylor's theorem gives, for every  $h$  satisfying the bound  $\|h\| \leq \delta$ , that

$$f(x_k + h) = f(x_k) + \langle \nabla_x f(x_k), h \rangle + \frac{1}{2} \langle h, \nabla_{xx} f(x_k)h \rangle + \epsilon_f$$

and

$$m_k(x_k + h) = f(x_k) + \langle g_k, h \rangle + \frac{1}{2} \langle h, \nabla_{xx} m_k(x_k)h \rangle + \epsilon_m,$$

where

$$|\epsilon_f| \leq \kappa_{\text{uft}} \|h\|^3 \quad \text{and} \quad |\epsilon_m| \leq \kappa_{\text{umt}} \|h\|^3. \quad (9.3.3)$$

Taking the difference of these two equations, we deduce that

$$\begin{aligned} \langle h, [\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)]h \rangle &= 2[f(x_k + h) - m_k(x_k + h) \\ &\quad - \langle \nabla_x f(x_k) - g_k, h \rangle - \epsilon_f + \epsilon_m] \end{aligned}$$

and therefore, using (9.3.1), AF.3, AM.4, the Cauchy–Schwarz inequality, (9.3.3), and Theorem 9.1.1,<sup>143</sup> that

$$\begin{aligned} |\langle h, [\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)]h \rangle| &\leq 2[(\kappa_{\text{cnd}} + \kappa_{\text{uft}} + \kappa_{\text{umt}})\delta^3 \\ &\quad + |\langle \nabla_x f(x_k) - g_k, h \rangle|] \\ &\leq 2(\kappa_{\text{cnd}} + \kappa_{\text{uft}} + \kappa_{\text{umt}} + \kappa_{\text{aeg}})\delta^3. \end{aligned}$$

Choosing  $h$  to be  $\delta$  times the normalized eigenvector corresponding to the largest eigenvalue of  $[\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)]$  in absolute value, we obtain that

$$\|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| \leq 2(\kappa_{\text{cnd}} + \kappa_{\text{uft}} + \kappa_{\text{umt}} + \kappa_{\text{aeg}})\delta.$$

This is (9.3.2) with  $\kappa_{\text{aeh}} = 2(\kappa_{\text{cnd}} + \kappa_{\text{uft}} + \kappa_{\text{umt}} + \kappa_{\text{aeg}})$ . □

This result has the following important consequence.

**Corollary 9.3.2** Suppose AF.1c and AF.4 hold together with AM.1c and AM.9. Suppose furthermore that the requirement of model validity is replaced by that of model second-order validity in Step 1 of Algorithm 9.1.1. Then AM.5 holds; that is,

$$\lim_{k \rightarrow \infty} \|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| = 0 \quad \text{whenever} \quad \lim_{k \rightarrow \infty} \|g_k\| = 0.$$

**Proof.** The mechanism of Step 1 of Algorithm 9.1.1 ensures that  $m_k$  is second-order valid in  $\mathcal{Q}_k(\delta_k)$  when  $\|g_k\|$  tends to zero for some  $\delta_k \in (0, \mu\|g_k\|]$ . Now, applying Theorem 9.3.1 gives the desired result. □

---

<sup>143</sup>With  $\delta$  replaced by  $\delta^{\frac{3}{2}}$ .

As a consequence, we may hope to apply the second-order convergence results of Sections 6.5 and 6.6, provided we replace validity by second-order validity in Step 1, AF.1 by AF.1c, AM.1 by AM.1c, and additionally assume AF.4 and AM.7–AM.9. There is, however, an additional difficulty, which arises because of the possibility of an infinite loop within Step 1. If the iterate  $x_k$  happens to be a first-order critical point ( $\nabla_x f(x_k) = 0$ ), but  $\nabla_{xx} f(x_k)$  has a negative eigenvalue, we may never see it because we might never leave Step 1 to compute a step which would then exploit this negative curvature. As a result, the algorithm would then converge to  $x_k$ , which is not what we want. Fortunately, there is an easy modification to Step 1 that prevents this undesirable situation. We consider replacing Step 1 by the following modified mechanism.

**Algorithm 9.3.1: Modified Step 1 for Algorithm 9.1.1**

**Step 1: Final criticality test.** If  $\|g_k\| \leq \epsilon_c$ , test if  $m_k$  is second-order valid in  $\mathcal{Q}_k(\delta_k)$  for some  $\delta_k \leq \mu \max(\|g_k\|, -\lambda_{\min}[\nabla_{xx} m_k(x_k)])$ . If this is the case, go to Step 2. Otherwise, perform as many improvement steps as necessary to ensure that  $m_k$  is second-order valid in  $\mathcal{Q}_k(\alpha \mu \max(\|g_k\|, -\lambda_{\min}[\nabla_{xx} m_k(x_k)]))$ .

Let us now examine the consequences of such a modification.

**Theorem 9.3.3** Suppose that AF.1c, AF.2–AF.4, AM.1c, AM.4, and AM.8–AM.9 hold. Suppose furthermore that an infinite loop occurs at iteration  $k$  of Algorithm 9.1.1 using the modified Step 1. Then  $x_k$  is second-order critical.

**Proof.** In order to prove the result, we again consider the sequence of successive models  $\{m_k^{(i)}\}_{i=0}^{\infty}$  defined in the proof of Lemma 9.1.2(iii). Since the loop within the modified Step 1 is infinite, the mechanism of this step implies in particular that, for all  $i > 0$ ,

$$\max(\|g_k^{(i)}\|, -\lambda_{\min}[\nabla_{xx} m_k^{(i)}(x_k)]) \leq \alpha^i \max(\|g_k\|, -\lambda_{\min}[\nabla_{xx} m_k(x_k)]) \stackrel{\text{def}}{=} \alpha^i \epsilon_k,$$

as in the proof of this lemma. This in turn yields that

$$-\lambda_{\min}[\nabla_{xx} m_k^{(i)}(x_k)] \leq \alpha^i \epsilon_k$$

for all  $i > 0$ , and hence that

$$\lambda_{\min}[\nabla_{xx} m_k^{(i)}(x_k)] \geq 0. \quad (9.3.4)$$

We then continue the proof as in Lemma 9.1.2 to deduce that

$$\nabla_x f(x_k) = 0 \text{ and } \lim_{i \rightarrow \infty} g_k^{(i)} = 0. \quad (9.3.5)$$

Since  $m_k^{(i)}$  is second-order valid in  $\mathcal{Q}_k(\alpha^i \epsilon_k)$  for every  $i > 0$ , by construction, Corollary 9.3.2 and the second part of (9.3.5) then imply that  $\nabla_{xx}m_k^{(i)}(x_k)$  converges to  $\nabla_{xx}f(x_k)$  and, therefore, that

$$\lambda_{\min}[\nabla_{xx}f(x_k)] \geq 0$$

because of (9.3.4). Combining this last inequality with the first part of (9.3.5) then ensures that  $x_k$  is second-order critical, as desired.  $\square$

Of course, if no infinite loop occurs within Step 1, the theory of Sections 6.5 and 6.6 applies under the extended assumptions discussed above, and we obtain convergence to second-order critical points from the results therein.

## 9.4 Conditional Models and Derivative-Free Optimization

### 9.4.1 Derivative-Free Minimization: Why and How?

Our first application of Algorithm 9.1.1 is in the context of derivative-free unconstrained minimization, that is, unconstrained problems where we suppose that the first (and a fortiori second) derivatives of the objective function cannot be computed for any  $x$ , although we assume they exist. The main motivation for examining algorithmic solutions to this problem is the high demand from practitioners for such tools. In typical cases,  $f(x)$  is *very* expensive to compute, and its derivatives are not available either because  $f(x)$  results from some physical, chemical, or econometric measure, or, more commonly, because it is the result of a possibly very large computer simulation, for which the source code is effectively unavailable. The occurrence of problems of this nature appears to be surprisingly frequent in the industrial world.

We consider here a possible approach to solving problems of this type, which we call *derivative-free optimization*, and in which we do not attempt to compute approximations to the unavailable derivative information directly but rather to derive an improvement in the objective function from a model of this function. This idea seems particularly attractive in that one can replace an expensive function evaluation by one of a much cheaper surrogate model. In this case, and especially when the problem is very complicated, it is possible to make considerable progress towards the solution at moderate cost.

The central idea is to use the available objective function values  $f(x_i)$  to construct a multivariate interpolation model for the objective function. This model is assumed to be valid in a trust-region neighbourhood of the current iterate. The interpolation model is then minimized within this trust region, yielding, we hope, a point with a low function value. As the algorithm proceeds and more objective function values become available, the points defining the interpolation model, the *interpolation set*, are updated in a way that attempts to preserve certain geometrical properties, and the trust-region radius is also adapted.

The first main ingredient of this algorithm is thus the choice of an adequate objective function model. We will here restrict ourselves to quadratic models of the form

$$m_k(x_k + s) = f(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle \quad (9.4.1)$$

for some  $g_k \in \mathbb{R}^n$  and some symmetric  $n \times n$  matrix  $H_k$  (see (6.1.7) [p. 117]) and also assume that  $\|\cdot\|_k$  is the  $\ell_2$  norm for all  $k$ . As indicated above,  $g_k$  and  $H_k$  will not be determined by the (possibly approximate) first and second derivatives of  $f(\cdot)$ , but rather by requiring that the model (9.4.1) interpolate function values at past points. Thus we will require that

$$m_k(y) = f(y) \text{ for all } y \in Y, \quad (9.4.2)$$

where the interpolation set  $Y$  is a set containing the current iterate  $x_k$  and points  $y$  such that  $f(y)$  is known for all  $y \in Y$ . Note that the cardinality of  $Y$  must be at least

$$p = 1 + n + \frac{1}{2}n(n+1) = \frac{1}{2}(n+1)(n+2)$$

to ensure that the quadratic model is entirely determined by the equations (9.4.2). However, if  $n > 1$ , this last condition is not sufficient to guarantee the existence or uniqueness of an interpolant. For instance, six points on a line do not determine a two-dimensional quadratic. Similarly, six interpolation points on a circle in the plane do not either, because any quadratic that is a multiple of the equation of the circle can be added to the interpolant without affecting (9.4.2). One therefore sees that some *geometric conditions* on  $Y$  must be added to the conditions (9.4.2) to ensure existence and uniqueness of the quadratic interpolant, which is essential for the validity of the model  $m_k$ . In the case of our second example, we must have that the interpolation points do not lie on any quadratic surface in  $\mathbb{R}^n$  or that the chosen model includes terms of degree higher than 2. More formally, we need a condition, known as *poisedness*, which relates directly to the interpolation points and the approximation space. If we choose a basis  $\{\phi_i(\cdot)\}_{i=1}^p$  of the linear space of  $n$ -dimensional quadratics, we may then express our model as a linear combination of these basis vectors, that is,

$$m_k(x) = \sum_{i=1}^p \alpha_i \phi_i(x).$$

Writing the conditions (9.4.2), we obtain that

$$f(y) = \sum_{i=1}^p \alpha_i \phi_i(y) \text{ for all } y \in Y. \quad (9.4.3)$$

This is a linear system in the coefficients  $\{\alpha_i\}_{i=1}^p$ , which is nonsingular if and only if the “interpolation determinant”

$$\partial(Y) = \det \begin{pmatrix} \phi_1(y_1) & \cdots & \phi_p(y_1) \\ \vdots & & \vdots \\ \phi_1(y_p) & \cdots & \phi_p(y_p) \end{pmatrix}$$

is nonzero. If this is the case, we say that  $Y = \{y_1, \dots, y_p\}$  is *poised*. Of course, for the model  $m_k$  to be valid in the sense discussed above, we will have to require that the interpolation set be poised, but we will also have to complete this definition by adding a condition on the distance between the current iterate  $x_k$  and the other points in  $Y$ . Furthermore, it turns out that we will not actually use the value of  $\partial(Y)$  directly, but will instead verify poisedness in a less direct manner.

#### 9.4.2 Basic Concepts in Multivariate Interpolation

In order to continue the discussion, we need to explore multivariate interpolation techniques a little further, which we do by considering the more general problem of finding interpolating polynomials of degree  $d$ . As above, we first choose a basis of the space of polynomials of degree  $d$  to initiate the process. For the case  $d = 2$ , corresponding to quadratic polynomials, we may, for instance, choose the natural basis given by

$$1, \{x_i\}_{1 \leq i \leq n}, \{x_i^2\}_{1 \leq i \leq n}, \{x_i x_j\}_{1 \leq i < j \leq n}.$$

In this framework, the points of the interpolation set  $Y$  are organized into  $d + 1$  *blocks*

$$Y^{[\ell]} = \{y_1^{[\ell]}, \dots, y_{|Y^{[\ell]}|}^{[\ell]}\} \quad (\ell = 0, \dots, d),$$

with the  $\ell$ th block containing

$$|Y^{[\ell]}| = \binom{\ell + n - 1}{\ell}$$

points, where the right-hand side is a binomial coefficient. To each point  $y_i^{[\ell]} \in Y^{[\ell]}$  corresponds a single *Newton fundamental polynomial* of degree  $\ell$  satisfying conditions

$$N_i^{[\ell]}(y_j^{[m]}) = \delta_{ij} \delta_{\ell m} \text{ for all } y_j^{[m]} \in Y^{[m]} \text{ with } m \leq \ell. \quad (9.4.4)$$

Conditions of this type are well known in univariate Newton interpolation. Other choices of bases are possible.<sup>144</sup>

The details of a procedure for constructing the basis of fundamental Newton polynomials for any given interpolation set  $Y$  are given in Algorithm 9.4.1. From a more abstract point of view, this algorithm may be considered as a version of the Gram–Schmidt orthogonalization procedure applied to the initial polynomial basis with respect to the inner product

$$\langle P, Q \rangle = \sum_{y \in Y} P(y)Q(y).$$

In this algorithm, we refer to the values  $N_i^{[\ell]}(y_i^{[\ell]})$  (the denominators in (9.4.5)) as interpolation *pivots*. In practice we need sufficiently large pivots. We may thus modify

---

<sup>144</sup>For instance, one may consider the Lagrange fundamental polynomials that are defined by the relations  $L_i(y_j) = \delta_{ij}$  for  $y_j \in Y$ .

**Algorithm 9.4.1: Finding the Newton fundamental polynomials**

**Step 0: Initialization.** Set the  $N_i^{[\ell]}$  ( $i = 1, \dots, |Y^{[\ell]}|, \ell = 0, \dots, d$ ) to the chosen polynomial basis. Set  $Y_{temp} = \emptyset$ .

**Step 1: Loop over the polynomials.** For  $\ell = 0, \dots, d$  and  $i = 1, \dots, |Y^{[\ell]}|$ ,

- choose some  $y_i^{[\ell]} \in Y \setminus Y_{temp}$  such that  $|N_i^{[\ell]}(y_i^{[\ell]})| \neq 0$ ;
- if no such  $y_i^{[\ell]}$  exists in  $Y \setminus Y_{temp}$ , reset  $Y = Y_{temp}$  and stop prematurely (the basis of Newton polynomials is incomplete);
- update the effective interpolation set by

$$Y_{temp} \leftarrow Y_{temp} \cup \{y_i^{[\ell]}\};$$

- normalize the current polynomial by

$$N_i^{[\ell]}(x) \leftarrow N_i^{[\ell]}(x) / |N_i^{[\ell]}(y_i^{[\ell]})|; \quad (9.4.5)$$

- update all Newton polynomials in block  $\ell$  and above by

$$N_j^{[\ell]}(x) \leftarrow N_j^{[\ell]}(x) - N_j^{[\ell]}(y_i^{[\ell]}) N_i^{[\ell]}(x) \quad (9.4.6)$$

for  $j \neq i, j = 1, \dots, |Y^{[\ell]}|$ , and

$$N_j^{[k]}(x) \leftarrow N_j^{[k]}(x) - N_j^{[k]}(y_i^{[\ell]}) N_i^{[\ell]}(x)$$

for  $j = 1, \dots, |Y^{[k]}|, k = \ell + 1, \dots, d$ .

the algorithm slightly by requiring that

$$|N_i^{[\ell]}(y_i^{[\ell]})| \geq \theta \quad (9.4.7)$$

for some  $\theta > 0$  instead of merely verifying that  $|N_i^{[\ell]}(y_i^{[\ell]})| \neq 0$ . In this case, we say that the Newton fundamental polynomials are “well poised” and call  $\theta$  the *pivoting threshold*.

Let us consider a small illustrative example of the application of Algorithm 9.4.1. Consider quadratic interpolation on a regular grid in the plane. We then require six interpolation points using three blocks ( $d = 2$ )

$$Y^{[0]} = \{(0, 0)\}, \quad Y^{[1]} = \{(1, 0), (0, 1)\}, \quad \text{and} \quad Y^{[2]} = \{(2, 0), (1, 1), (0, 2)\},$$

and use the basis functions 1,  $x_1$ ,  $x_2$ ,  $x_1^2$ ,  $x_1 x_2$ , and  $x_2^2$ . Applying Algorithm 9.4.1 then yields the polynomials

$$N_1^{[0]} = 1, \quad N_1^{[1]} = x_1, \quad N_2^{[1]} = x_2,$$

$$N_1^{[2]} = \frac{1}{2}(x_1^2 - x_1), \quad N_2^{[2]} = x_1 x_2, \quad \text{and} \quad N_3^{[2]} = \frac{1}{2}(x_2^2 - x_2),$$

for which the conditions (9.4.4) may easily be verified. We also illustrate the application of the Newton fundamental polynomials by a graphical example. The problem is to interpolate the function

$$f(x_1, x_2) = (x_2 - 3 \sin x_1)^2 - x_1^2 + \frac{1}{100}(x_1 - 4)^4 + e^{2x_1 - 8}$$

on the interpolation set  $Y$  whose block structure is

$$Y^{[0]} = \{(0, 0)\}, \quad Y^{[1]} = \{(-1, -1), (\frac{1}{2}, -1)\}, \quad \text{and} \quad Y^{[2]} = \{(-\frac{1}{2}, \frac{3}{2}), (1, 1), (-\frac{3}{2}, \frac{1}{2})\}.$$

The level curves of this function are shown in the top-left picture of Figure 9.4.1 together with the points in  $Y$  (indicated by small circles). The level curves of the Newton fundamental polynomials  $N_1^{[1]}$ ,  $N_2^{[1]}$ ,  $N_1^{[2]}$ ,  $N_2^{[2]}$ , and  $N_3^{[2]}$  are also shown (from left to right and top to bottom). The level curves of  $N_1^{[0]}$  are not shown, as this polynomial is identically equal to 1.

Notice that after applying Algorithm 9.4.1,  $Y$  is always poised since we only include the points that create nonzero pivots. This is true even if the procedure stops prematurely with an incomplete basis of Newton polynomials, which then results in an interpolating polynomial that is not of full degree  $d$  (meaning that it does not include contributions of all polynomials of the chosen basis; see the second part of Step 1 of the algorithm).

Observe also that, if Algorithm 9.4.1 stops prematurely, then  $N_i^{[\ell]}(y) = 0$  for all vectors  $y$  of the original interpolation set that have not been included in the restricted interpolation set computed by the algorithm. This is the equation of a surface of which any multiple can be added to the interpolant without affecting the interpolation conditions (9.4.2), as was the equation of the circle in our example above. Hence, although the strict subset produced by the algorithm is poised, the original set  $Y$  cannot be.

If we assume that we have computed the Newton fundamental polynomials by applying Algorithm 9.4.1, we may now show how a polynomial interpolant can be built using them. As is the case for univariate Newton interpolation, this is done using finite differences, but this notion must be generalized to the multivariate case. For a given  $x \in \mathbb{R}^n$ , we define the generalized finite differences  $\lambda_\ell(x)$  recursively by the formulae

$$\lambda_0(x) \stackrel{\text{def}}{=} f(x) \quad \text{and} \quad \lambda_{\ell+1}(x) \stackrel{\text{def}}{=} \lambda_\ell(x) - \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(x) \quad (\ell = 0, \dots, d-1). \quad (9.4.8)$$

Note that  $\lambda_{\ell+1}(x)$  is defined if and only if all the  $N_i^{[j]}(x)$  are defined for  $j = 0, \dots, \ell$  and  $i = 1, \dots, |Y^{[j]}|$ .

It is then possible to use these generalized finite differences to build our interpolation polynomial, as is shown in Theorem 9.4.1.

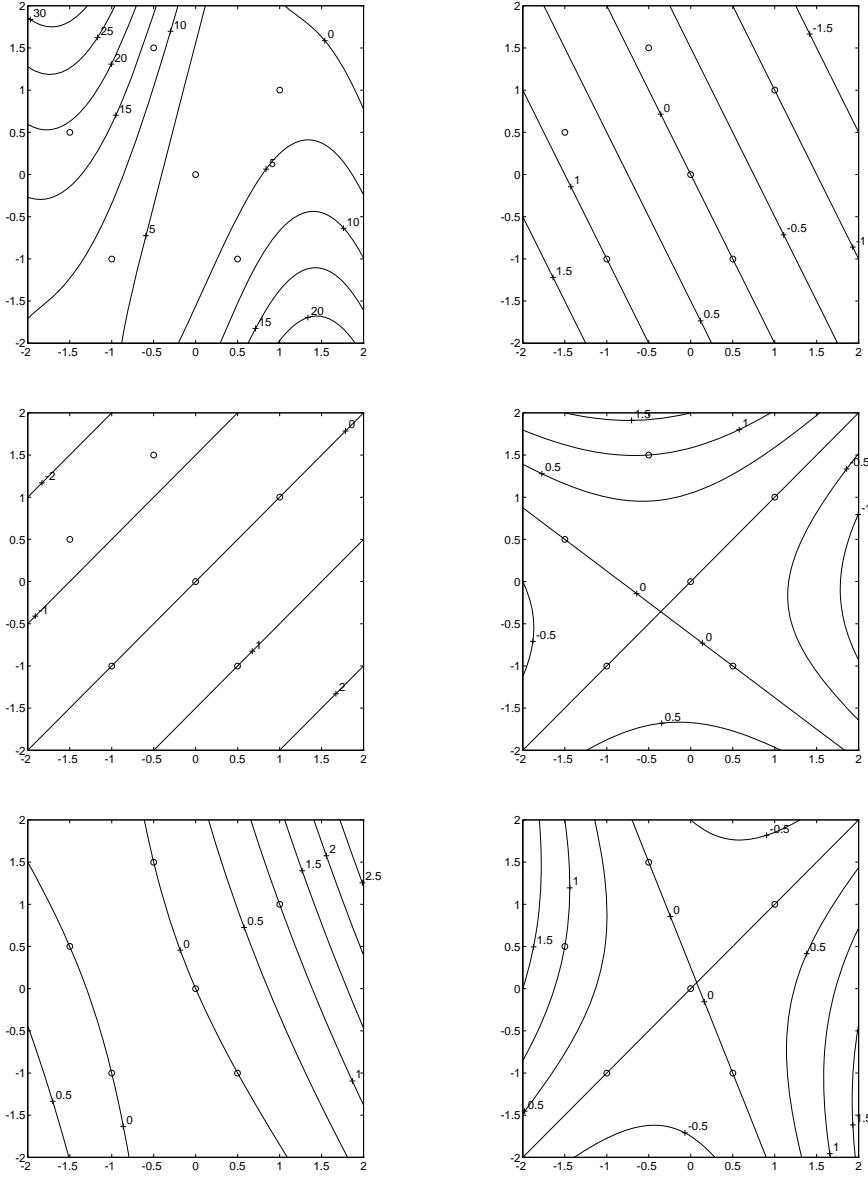


Figure 9.4.1: The function to interpolate and the corresponding Newton fundamental polynomials  $N_1^{[1]}$ ,  $N_2^{[1]}$ ,  $N_1^{[2]}$ ,  $N_2^{[2]}$ , and  $N_3^{[2]}$  (from left to right and top to bottom).

**Theorem 9.4.1** Suppose that the Newton fundamental polynomials  $N_i^{[\ell]}(x)$  are defined for  $\ell = 0, \dots, d$  and  $i = 1, \dots, |Y^{[\ell]}|$ . Then

$$m_k(x) = \sum_{\ell=0}^d \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(x) \quad (9.4.9)$$

is well defined and satisfies the interpolation conditions (9.4.2).

**Proof.** We have that, for each  $j = 0, \dots, d$ ,

$$\sum_{i=1}^{|Y^{[j]}|} \lambda_j(y_i^{[j]}) N_i^{[j]}(y_k^{[j]}) = \lambda_j(y_k^{[j]}) \quad (9.4.10)$$

because of (9.4.4). Moreover, using (9.4.8) repeatedly, we deduce that

$$\begin{aligned} \lambda_j(y_k^{[j]}) &= \lambda_{j-1}(y_k^{[j]}) - \sum_{i=1}^{|Y^{[j-1]}|} \lambda_{j-1}(y_i^{[j-1]}) N_i^{[j-1]}(y_k^{[j]}) \\ &= \lambda_{j-2}(y_k^{[j]}) - \sum_{\ell=j-2}^{j-1} \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(y_k^{[j]}) \\ &= \lambda_0(y_k^{[j]}) - \sum_{\ell=0}^{j-1} \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(y_k^{[j]}) \\ &= f(y_k^{[j]}) - \sum_{\ell=0}^{j-1} \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(y_k^{[j]}). \end{aligned} \quad (9.4.11)$$

Combining (9.4.10) and (9.4.11), we obtain that, for all  $j = 0, \dots, d$ ,

$$f(y_k^{[j]}) = \sum_{\ell=0}^j \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(y_k^{[j]}) = \sum_{\ell=0}^d \sum_{i=1}^{|Y^{[\ell]}|} \lambda_\ell(y_i^{[\ell]}) N_i^{[\ell]}(y_k^{[j]}), \quad (9.4.12)$$

where we have used (9.4.4) again to derive the last equality. We therefore conclude from (9.4.12) that (9.4.2) holds and that  $m_k$  is well defined, because the required Newton fundamental polynomials are, by assumption, all well defined.  $\square$

We return to our graphical example of Figure 9.4.1 and show, in Figure 9.4.2, the level curves of the interpolation functions defined by (9.4.9) corresponding to the subsets

$$\begin{aligned} &\{(0, 0), (-1, -1), (\frac{1}{2}, -1)\}, \quad \{(0, 0), (-1, -1), (\frac{1}{2}, -1), (-\frac{1}{2}, \frac{3}{2})\}, \\ &\{(0, 0), (-1, -1), (\frac{1}{2}, -1), (-\frac{1}{2}, \frac{3}{2}), (1, 1)\}, \text{ and} \\ &\{(0, 0), (-1, -1), (\frac{1}{2}, -1), (-\frac{1}{2}, \frac{3}{2}), (1, 1), (-\frac{3}{2}, \frac{1}{2})\}. \end{aligned}$$

These subsets are all poised and result in different interpolating polynomials, the first being linear and the last three quadratic.

In principle, Algorithm 9.4.1, coupled with the definition of the generalized finite differences in (9.4.8) and the formula (9.4.9), is all we need to build suitable models of the objective function. However, the calculation of the  $\lambda_\ell(y_i^{[\ell]})$  from (9.4.8) is too slow to be of practical use. Fortunately, it turns out that a more efficient procedure can be introduced, which is reminiscent of the classical triangular scheme for univariate interpolation. First define

$$\lambda_{i,\ell} = f(y_i^{[\ell]}) \quad (i = 1, \dots, |Y^{[\ell]}|, \ell = 0, \dots, d).$$

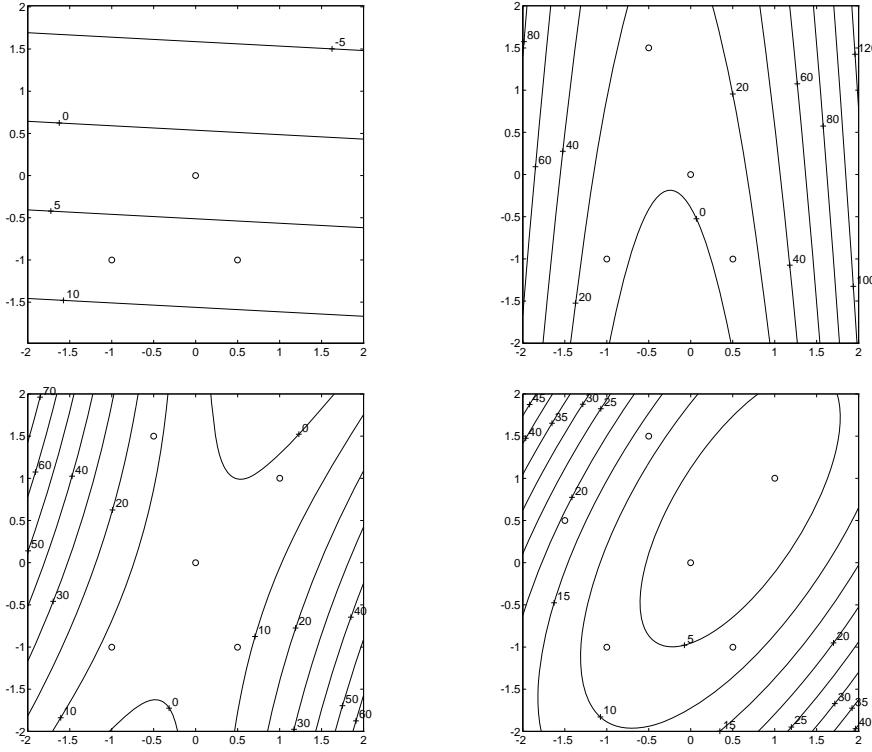


Figure 9.4.2: Level curves of interpolation polynomials of increasing degree, corresponding to subsets of the interpolation set  $Y$ . The level curves of the interpolated function are shown in the top-left picture of Figure 9.4.1.

We now describe an algorithm which progressively transforms these quantities into

$$\lambda_{i,\ell} = \lambda_\ell(y_i^{[\ell]}), \quad (9.4.13)$$

which are the generalized finite differences that we need (see (9.4.9)). Because of the definition (9.4.8), we observe that

$$\lambda_{1,0} = f(y_1^{[0]}) = \lambda_0(y_1^{[0]}),$$

and (9.4.13) already holds for  $\ell = 0$ . We may now update all the  $\lambda_{i,\ell}$  for  $i = 1, \dots, |Y^{[\ell]}|$  and  $\ell = 1, \dots, d$  by

$$\lambda_{i,\ell} = \lambda_{i,\ell} - \lambda_{1,0} N_1^{[0]}(y_i^{[\ell]}) = \lambda_0(y_i^{[\ell]}) - \lambda_0(y_1^{[0]}) N_1^{[0]}(y_i^{[\ell]}) = \lambda_1(y_i^{[\ell]}).$$

This implies that

$$\lambda_{i,1} = \lambda_1(y_i^{[1]}) \quad (i = 1, \dots, |Y^{[1]}|),$$

and thus (9.4.13) now holds for  $\ell = 0, 1$ . We then update all the  $\lambda_{i,\ell}$  for  $i = 1, \dots, |Y^{[\ell]}|$  and  $\ell = 2, \dots, d$ , and so on, each time ensuring (9.4.13) for one more block. By this means, we obtain Algorithm 9.4.2 for computing the generalized finite differences.

**Algorithm 9.4.2: Finding the generalized finite differences**

**Step 0: Initialization.** For  $i = 1, \dots, |Y^{[\ell]}|$  and  $\ell = 0, \dots, d$ , set  $\lambda_{i,\ell} = f(y_i^{[\ell]})$ .

**Step 1: Consider the blocks by increasing index.** For  $k = 1, \dots, d$  successively, compute

$$\lambda_{i,\ell} = \lambda_{i,\ell} - \sum_{j=1}^{|Y^{[k-1]}|} \lambda_{j,k-1} N_j^{[k-1]}(y_i^{[\ell]}) \quad (i = 1, \dots, |Y^{[\ell]}|, k \leq \ell \leq d).$$

Then return  $\lambda_\ell(y_i^{[\ell]}) = \lambda_{i,\ell}$  for  $i = 1, \dots, |Y^{[\ell]}|$  and  $\ell = 0, \dots, d$ .

Thus we conclude that we can determine the model  $m_k$  satisfying (9.4.2) from (9.4.9) provided Algorithm 9.4.1 terminates normally, since the generalized finite differences are always well defined by Algorithm 9.4.2 so long as the Newton fundamental polynomials exist. Furthermore, this conclusion does not depend on the values  $\{f(y_i^{[\ell]})\}$ . Returning to (9.4.3), that is, if Algorithm 9.4.1 terminates normally, then the linear map from  $\mathbb{R}^p$  (considered as the space of the  $\alpha_i$ ) to  $\mathbb{R}^p$  (considered as the space of the  $f(y)$ ) is surjective. Hence it must be one-to-one, which is in turn equivalent to  $\partial(Y) \neq 0$ . In view of our remark following the description of Algorithm 9.4.1, this leads us to the following conclusion.

**Theorem 9.4.2** The set  $Y$  is poised if and only if all pivots are nonzero in Algorithm 9.4.1.

Also observe that, if Algorithm 9.4.1 terminates prematurely, it still determines a restricted interpolation subset that is poised (by construction).

### 9.4.3 The Interpolation Error

It is remarkable that one can also derive a bound, based on the theory presented above, on the distance between  $f$  and  $m_k$  at a point  $x$ . This bound depends upon the concept of a *path* between the zeroth block and  $x$ , which uses a sequence of points of  $Y$  of the form

$$\pi(x) = (z_0 = y_0^{[0]}, z_1, \dots, z_d, z_{d+1} = x),$$

where  $z_i \in Y^{[i]}$  ( $i = 0, \dots, d$ ). A path therefore contains  $x$  and exactly one interpolation point from each block. Let us denote by  $\Pi(x) = \{\pi(x)\}$  the set of all possible paths from  $Y^{[0]} = \{y_0^{[0]}\} = \{z_0\}$  to  $x$ . Figure 9.4.3 shows the six paths of  $\Pi(-\frac{1}{2}, 0)$  for the two-dimensional example used above.

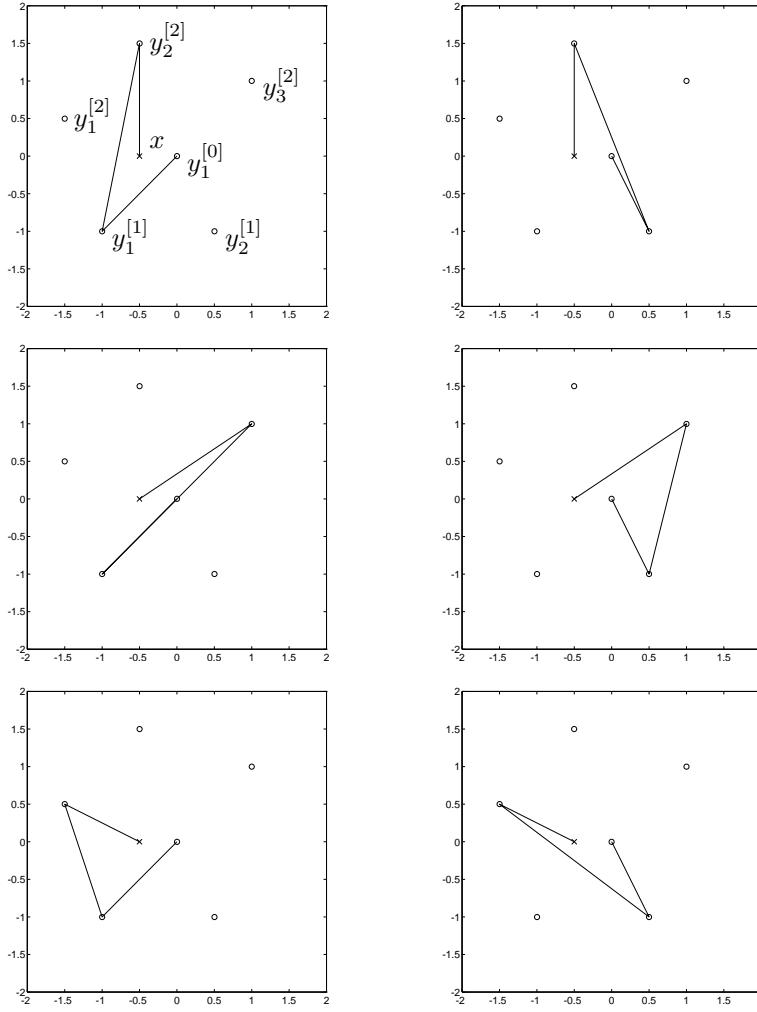


Figure 9.4.3: The six paths of  $\Pi(-\frac{1}{2}, 0)$  for the two-dimensional example of Figure 9.4.1. The points in  $Y$  are labelled in the first picture.

Using this notion, it is possible to derive<sup>145</sup> the inequality

$$|f(x) - m_k(x)| \leq \frac{n^{d+1} \|f^{(d)}\|_\infty}{(d+1)!} \sum_{\pi(x) \in \Pi(x)} \left[ \prod_{i=0}^d \|z_{i+1} - z_i\|_\infty |N_i^{[i]}(z_{i+1})| \right] \quad (9.4.14)$$

for all  $x$ , where  $f^{(d)}$  is the  $d$ th derivative tensor of  $f$ . Interestingly, the quantities  $N_i^{[i]}(z_{i+1})$  are all computed in the course of the evaluation of the generalized finite differences  $\lambda_i(y_i^{[\ell]})$ . We see that the error between  $m_k(x)$  and  $f(x)$  is small if we can make the values  $N_i^{[i]}(z_{i+1})\|z_{i+1} - z_i\|_\infty$  small. If all the interpolation points and the

<sup>145</sup>The proof of the bound is very technical and lengthy, and will be omitted. The reader is referred to Sauer and Xu (1995), Theorem 3.11, for a first step, which then can be simplified (Sauer, private communication, February 1996) to (9.4.14).

point  $x$  are chosen in a given hypersphere of radius  $\delta$ , it is then possible to provide an upper bound on the maximal error.

**Theorem 9.4.3** Suppose that an arbitrary  $x_k \in \mathbb{R}^n$  and a  $\delta > 0$  are given, together with the interpolation degree  $d$ . Then it is possible to construct an interpolation set  $Y$  containing  $x_k$  and yielding a complete basis of Newton polynomials such that all  $y \in Y$  satisfy

$$y \in \mathcal{Q}_k(\delta)$$

and also that

$$|N_j^{[\ell]}(x)| \leq \kappa_{\text{nfp}}$$

for all  $\ell = 0, \dots, d$ , all  $j = 1, \dots, |Y^{[\ell]}|$ , and all  $x \in \mathcal{Q}_k(\delta)$ , for some<sup>146</sup>  $\kappa_{\text{nfp}} > 1$ .

**Proof.** We consider Algorithm 9.4.1, but instead of choosing the interpolation points  $y_i^{[\ell]}$  from a given set  $Y$ , we determine them successively by

$$y_i^{[\ell]} = \arg \max_{x \in \mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)|. \quad (9.4.15)$$

Note that, because of the independence of the polynomials in the basis, none of the fundamental polynomials can become identically zero, and thus the value of the maximum in (9.4.15), and hence of the denominator in (9.4.5), is always nonzero. Applying this algorithm, we see that the normalization step ensures that

$$\max_{x \in \mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)| \leq 1, \quad (9.4.16)$$

except that this value may be modified when polynomials of the same block are updated by (9.4.6) later in the computation. Consider one such update: using (9.4.6) and (9.4.16), we have that

$$\begin{aligned} \max_{x \in \mathcal{Q}_k(\delta)} |\hat{N}_j^{[\ell]}(x)| &\leq \max_{x \in \mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)| + |N_j^{[\ell]}(y_i^{[\ell]})| \max_{x \in \mathcal{Q}_k(\delta)} |N_i^{[\ell]}(x)| \\ &\leq 2 \max_{x \in \mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)|, \end{aligned}$$

where  $\hat{N}_j^{[\ell]}$  denotes the  $j$ th polynomial of the  $\ell$ th level after the update (9.4.6). Thus, at the end of the process and taking into account that (9.4.16) holds for  $i = j$  when  $N_j^{[\ell]}(x)$  is first normalized,

$$\max_{x \in \mathcal{Q}_k(\delta)} |N_j^{[\ell]}(x)| \leq 2^{|Y^{[\ell]}|-j} \leq 2^{|Y^{[\ell]}|} \leq 2^{|Y^{[d]}|} \stackrel{\text{def}}{=} \kappa_{\text{nfp}},$$

as desired.  $\square$

---

<sup>146</sup> “nfp” stands for “Newton fundamental polynomials”.

Observe that  $\kappa_{\text{nfp}}$  is independent of  $x_k$  and  $\delta$ .

We may then combine (9.4.14) and Theorem 9.4.3 to obtain a result linking our problem of unconstrained derivative-free optimization with the framework of conditional models that we studied in the preceding paragraphs. However, this result requires the following additional assumption.

**AF.5** The gradient of  $f$  is uniformly bounded; that is, there exists a constant<sup>147</sup>  $\kappa_{\text{ufg}}$  such that

$$\|\nabla_x f(x)\| \leq \kappa_{\text{ufg}}$$

for all  $x \in \mathbb{R}^n$ .

Observe that this assumption is unnecessary if we assume that the iterates  $x_k$  remain in a bounded domain and if the trust-region radii are uniformly bounded, because of AF.1. Using this assumption, we may now establish the following crucial bound.

**Theorem 9.4.4** Suppose AF.1, AF.3, and AF.5 hold, and also that we are using a model of the form (9.4.9). Suppose furthermore that  $Y$  is poised and that

- (i) the model is at least fully linear, that is,

$$|Y| \geq n + 1; \quad (9.4.17)$$

- (ii) the points in  $Y$  are sufficiently close to  $x_k$ , that is,

$$y \in \mathcal{Q}_k(\delta) \text{ for all } y \in Y; \quad (9.4.18)$$

- (iii) the Newton fundamental polynomials are bounded, that is,

$$|N_i^{[\ell]}(y_j^{[\ell+1]})| \leq \kappa_{\text{nfp}} \quad (i = 1, \dots, |Y^{[\ell]}|, j = 1, \dots, |Y^{[\ell+1]}|, \ell = 0, \dots, d - 1) \quad (9.4.19)$$

and

$$|N_i^{[d]}(x)| \leq \kappa_{\text{nfp}} \quad (i = 1, \dots, |Y^{[d]}|, x \in \mathcal{Q}_k(\delta)). \quad (9.4.20)$$

Then

$$|f(x) - m_k(x)| \leq \kappa_{\text{cnd}} \max[\delta^2, \delta^3] \quad (9.4.21)$$

for all  $x \in \mathcal{Q}_k(\delta)$  and some constant  $\kappa_{\text{cnd}} > 0$  independent of  $k$ .

**Proof.** We consider the bound (9.4.14) for  $d = 1$  and  $d = 2$ . Because of AF.3 and AF.5, we obtain that the quantity  $\|f^{(d)}\|$  is bounded above by  $\max(\kappa_{\text{ufg}}, \kappa_{\text{ufh}})$ . Furthermore, (9.4.18) ensures that

$$\|y_i - x_k\| \leq \delta \quad (i = 1, \dots, p),$$

<sup>147</sup>“ufg” stands for “upper bound on the function’s gradient”.

and therefore that

$$\|y_i - y_j\| \leq 2\delta \quad (i, j = 1, \dots, p, i \neq j).$$

Combining these bounds with (9.4.14) for  $d = 1$  and  $d = 2$  then yields that

$$|f(x) - m_k(x)| \leq 2n^3(n+1)\delta^2\kappa_{\text{nfp}}^2 \max[\kappa_{\text{ufg}}, \frac{2}{3}n\delta\kappa_{\text{nfp}}\kappa_{\text{ufh}}],$$

which obviously gives (9.4.21) with  $\kappa_{\text{end}} \stackrel{\text{def}}{=} 2n^4(n+1) \max(\kappa_{\text{ufg}}, \kappa_{\text{ufh}})\kappa_{\text{nfp}}^3$ .  $\square$

Note that we may have to restrict our attention to the interpolation set  $Y$  resulting from the application of Algorithm 9.4.1 if the original set  $Y$  is not poised. Note also that the use of  $\kappa_{\text{nfp}}$  in (9.4.19) and (9.4.20) is coherent in view of Theorem 9.4.3. Finally, we observe that AF.5 would not be needed if, instead of (9.4.17), we had required that

$$|Y| = p,$$

which is to say that the model is fully quadratic. Indeed, we would only consider the case  $d = 2$  in the proof of Theorem 9.4.4, and AF.3 would therefore suffice.

If we return once more to our example of Figures 9.4.1 and 9.4.2, we observe that all conditions of Theorem 9.4.4 are satisfied by each interpolant of Figure 9.4.2 for  $x_k = 0$  and  $\delta = 2$ .

The value of  $\kappa_{\text{nfp}}$  is very large, except for small values of  $n$  and  $d$ , and only serves a theoretical purpose. In practice the threshold pivoting strategy<sup>148</sup> applied to Algorithm 9.4.1 may be a better way to verify the adequacy of the geometry, because as long as the new point satisfies the threshold provision we can include it. The use of the procedure suggested in the proof of Theorem 9.4.3 may thus be viewed as a (very) last resort if threshold pivoting rules do not prevent the  $|N_j^{[\ell]}(y_j^{[\ell+1]})|$  from becoming very large. Note that, because we consider models that are at most quadratic, the solution of the global maximization subproblems arising in (9.4.15) may be obtained by applying Algorithm 7.3.4.

The connection with the framework of conditional models is now easy to establish. In view of Theorem 9.4.4, we define the model  $m_k$  to be valid in  $\mathcal{Q}_k(\delta)$  if and only if the associated interpolation set  $Y$  satisfies conditions (9.4.17)–(9.4.20). If we wish to check this property, for any iteration  $k$ , it is easy to verify (9.4.17) and (9.4.18) directly. Furthermore, the values of  $|N_i^{[\ell]}(y_j^{[\ell+1]})|$  are computed by Algorithm 9.4.2 and may therefore simply be compared to an a priori given bound like  $\kappa_{\text{nfp}}$ . Finally, (9.4.20) is more expensive to verify, as it requires the solution of the  $|Y^{[d]}|$  problems

$$\max_{x \in \mathcal{B}_k} |N_i^{[d]}(x)| \quad (i = 1, \dots, |Y^{[d]}|),$$

but this is possible and acceptable, especially if the cost of evaluating the objective function is high. As a consequence, AM.7 is satisfied. We also deduce from the proof of Theorem 9.4.3 that AM.8 is fulfilled, because it is enough to apply the procedure

---

<sup>148</sup>That is, insisting on (9.4.7).

described in this proof to obtain a suitable interpolation set  $Y$  and, therefore, a valid model. It therefore defines a perfectly acceptable improvement procedure, as seen in the context of conditional models. In practice, however, one may instead rely on the pivot threshold technique to verify that the model is valid, at the risk of losing our theoretical guarantee, as explained in the previous paragraph. Many different techniques are then possible. For instance, a reasonable strategy is to remove a point  $y_- = y_i^{[\ell]} \neq x_k$  from  $Y$  and to replace it by

$$y_+ = \arg \max_{x \in \mathcal{B}_k} |N_i^{[\ell]}(x)|.$$

Priority should be given to those points not in  $\mathcal{B}_k$ , if any. Theorem 9.4.3 then ensures that (9.4.19) and (9.4.20) will hold after a finite number of such replacements. A computationally expensive version of the improvement procedure would compute  $y_+$  for every possible choice of  $y_-$ , and select the one for which  $|N_i^{[\ell]}(y_+)|$  is maximal, but substantially cheaper versions can be designed by choosing  $y_-$  using the information available from existing interpolation model calculations. For instance, one may consider the vectors  $y_i^{[\ell]}$  corresponding to polynomials for which  $|N_j^{[\ell-1]}(y_i^{[\ell]})|$  is large for some  $j$ , or one may choose to replace interpolation points corresponding to small pivots in Algorithm 9.4.1.<sup>149</sup> We also observe at this point that Theorem 9.4.4 can be modified to ensure second-order validity of the model: it is enough to replace (9.4.17) by the condition that

$$|Y| = p.$$

In other words, the model is second-order valid if it is fully quadratic and satisfies conditions (9.4.18)–(9.4.20), which can be checked and enforced just as for (9.4.17)–(9.4.20).

To complete our discussion, we still need to clarify one more algorithmic issue, concerning the inclusion of  $\hat{x}_k$  in  $Y$  whenever  $x_{k+1} = \hat{x}_k$ . If  $|Y| < p$ , we may simply add  $\hat{x}_k$  to  $Y$ . However, if  $|Y| = p$ , we need to remove a point  $y_-$  from  $Y$ . Ideally, this point should be chosen to make the geometry of  $Y$  as good as possible. There are various ways to attempt to achieve this goal. For instance, one might choose to remove the  $y_- = y_i^{[\ell]}$  for which  $|N_i^{[\ell]}(y_-)|$  is largest, thereby trying to make the pivots as large as possible, but other techniques are also possible.

As a result of our discussion, we see that trust-region algorithms for unconstrained derivative-free minimization may be designed that lie within the class of algorithms using conditional models. The convergence theory developed for this class then fully applies.

## Notes and References for Section 9.4

The approach we have developed for derivative-free optimization has its origin in the little-known work of Winfield (1969, 1973), who proposed using available objective function values

---

<sup>149</sup>A closer look at the mechanism of this algorithm furthermore indicates that significant computational savings can be achieved if the point to be replaced is selected in  $Y^{[d]}$ , or at least in the blocks of higher index, whenever possible.

$f(x_i)$  to build a quadratic model by interpolation. This model is assumed to be valid in a neighbourhood of the current iterate, which is described as a trust region whose radius is iteratively adjusted. The model is then minimized within the trust region, hopefully yielding a point with a low function value. As the algorithm proceeds and more objective function values become available, the set of points defining the interpolation model is updated in such a way that it always contains the points closest to the current iterate. This idea was resurrected by Powell (1994a), who proposed a method for constrained optimization in which the objective function and constraints are approximated by linear multivariate interpolation. Exploring the idea further, Powell (1994b) then described an algorithm for unconstrained optimization using a multivariate quadratic interpolation model of the objective function in a trust-region framework, an approach extremely similar to that of Winfield, although seemingly independent. The crucial difference between Powell's and Winfield's proposals is that the set of interpolation points in the former is updated in a way that preserves its geometrical properties. This is achieved by exploiting a relation between the ratio of successive values of  $\partial(Y)$  and the Lagrange fundamental polynomials associated with the interpolation problem, these polynomials also being used to construct the interpolation model itself. Powell (1998b) showed that their coefficients can be updated from iteration to iteration in a numerically stable way.

A variant of this quadratic interpolation scheme using Newton fundamental polynomials was then discussed in Conn and Toint (1996), where encouraging numerical results were presented. Powell (1996) subsequently revisited the idea of using a model derived from multivariate quadratic interpolation and showed computational results similar to those of Conn and Toint. The first global convergence theorems (to first-order critical points) for methods of this type were presented by Conn, Scheinberg, and Toint (1997a), together with a description of alternative techniques to enforce the desired geometrical properties of the set of interpolation points. These techniques are based on the exploitation of the theory of multivariate Newton interpolation developed in De Boor and Ron (1992), Sauer and Xu (1995), and Sauer (1995 and private communication, February 1996). We have followed this approach in our exposition of the techniques for derivative-free optimization. The extension of second-order convergence results to these methods is new. A survey covering techniques of this type can be found in Conn, Scheinberg, and Toint (1997b), while a wider coverage of methods of optimization without derivatives can be found in Powell (1998a). See also Conn, Scheinberg, and Toint (1998). We also note that the suggestion of moving to points at which the objective function is evaluated solely for interpolation purposes (as is allowed by Step 5 of Algorithm 9.1.1) was made by Mifflin (1975b), who suggested a linesearch algorithm for derivative-free optimization based on finite differences.

## 9.5 Conditional Models and Models with Memory

### 9.5.1 Adding Memory to the Model of the Objective Function

We now consider a second application of the framework using conditional models. The motivation for this application is that the model of the objective function is often too local, in the sense that although it guarantees that progress can be made locally,

it does not necessarily reflect the overall shape of the objective function. In other words, a trust-region algorithm whose model  $m_k$  is solely based on the values of the first and second derivatives at the current iterate  $x_k$  may sometimes be trapped into exploring “local wriggles” of the objective function, instead of taking a broader view of the overall shape of that function. This is especially true for the commonly occurring case where the model is derived from a Taylor series approximation to the objective. In this subsection, we investigate a possible definition of the model that is a compromise between maintaining a global view of the objective and guaranteeing local progress.

If we wish to obtain information about the global shape of the objective, we may do so by trying to “remember” the models we have seen in previous iterations, and mixing this information with the local model at  $x_k$ . If we call the local model  $q_k(x)$ ,<sup>150</sup> we consider using the model

$$m_k(x) = (1 - \alpha_k)q_k(x) + \alpha_k m_{k-1}(x), \quad (9.5.1)$$

where  $\alpha_k$  is a parameter that we may choose at each iteration between 0 and some fixed upper bound  $\bar{\alpha} \in [0, 1]$ . This model thus includes some “memory” of the history of previous models, in the hope that this will better reflect the global behaviour of the objective. The amount of memory used at iteration  $k$  is controlled by  $\alpha_k$ , and if  $\alpha_k = 0$  we have no memory of the past and we recover the usual entirely local model  $m_k = q_k$ . It is therefore natural to ask how this memory parameter should be chosen at each iteration. Note that one expects a local model to be more useful when the length of the step is short. Thus it seems reasonable to require that  $\alpha_k$  must be of the order of  $\|s_{k-1}\|$ . This consideration may be expressed by a condition of the following type.

**AM.10** For each  $k$ , one has that

$$\alpha_k \leq \min[\bar{\alpha}, \kappa_\alpha \|s_{k-1}\|^\psi]$$

for some constant  $\bar{\alpha} \in [0, 1]$ , some  $\psi > 0$ , and some constant  $\kappa_\alpha > 0$  such that

$$\kappa_\alpha \left( \frac{\kappa_{\text{une}}}{\gamma_1} \right)^\psi \leq 1.$$

Furthermore,  $\alpha_0 = 0$ .

The justification of the requirement on  $\kappa_\alpha$  and  $\psi$  will become clear later. Note that the last part of this assumption simply states that there is nothing to remember at the first iteration. We also assume in this description that the local model  $q_k(x)$  is truly local and defined everywhere, a condition that we formulate as follows.

**AM.11** For each  $k$ ,  $q_k(x)$  is entirely determined by  $x_k$  and  $x$  and is defined for all  $x \in \mathbb{R}^n$ . Furthermore,  $q_k$  is twice-continuously differentiable,

$$q_k(x_k) = f(x_k), \quad \nabla_x f(x_k) = \nabla_x q_k(x_k), \quad \text{and} \quad \|\nabla_{xx} q_k(x)\| \leq \kappa_{\text{umh}}$$

for all  $k$  and all  $x \in \mathcal{B}_k$ .

---

<sup>150</sup>The local model  $q_k$  is typically given by a quadratic defined by the Taylor series of  $f$  at  $x_k$  truncated at second order, that is,  $q_k(x_k + s) = f(x_k) + \langle \nabla_x f(x_k), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} f(x_k) s \rangle$ .

Observe that AM.11 and (9.5.1) together imply that  $m_k(x)$  is always well defined for all  $k$  and all  $x \in \mathcal{B}_k \subset \mathbb{R}^n$ .

If the use of this model produces successful steps, we obtain the progress we are looking for. On the other hand, if the steps predicted by using (9.5.1) are not successful, and thus shrink to zero, AM.10 and the definition (9.5.1) together ensure that  $m_k$  approaches  $q_k$ , the local model for which we know that progress is ultimately possible provided  $x_k$  is not first-order critical. Hence a successful step will eventually occur, and the algorithm cannot get trapped at a noncritical point. The purpose of the following section is to put this informal outline on a sound theoretical footing.

### 9.5.2 The Effect of Memory on the Modelling Error

We first observe that the definition (9.5.1) gives that

$$\begin{aligned} m_k(x) &= (1 - \alpha_k)q_k(x) + \alpha_k m_{k-1}(x) \\ &= (1 - \alpha_k)q_k(x) + \alpha_k(1 - \alpha_{k-1})q_{k-1}(x) + \alpha_k\alpha_{k-1}m_{k-2}(x) \\ &= \dots \\ &= \sum_{i=0}^k (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) q_i(x). \end{aligned} \tag{9.5.2}$$

If we now assume that the values of the past models  $q_i(x)$  remain bounded, the identity (9.5.2) allows us to derive the key relation between the local and global models.

**Theorem 9.5.1** Suppose that AM.11 holds. Suppose furthermore that iteration  $k$  is unsuccessful and that the last successful iteration before  $k$  is iteration  $t$ . Then there exists a constant<sup>151</sup>  $\kappa_{\text{mma}} > 0$  such that

$$|m_k(x) - q_k(x)| \leq \kappa_{\text{mma}} \kappa_k \alpha_{t,k}^{k-t} \tag{9.5.3}$$

for each  $x \in \mathcal{B}_k$ , where

$$\alpha_{t,k} \stackrel{\text{def}}{=} \max_{i=t+1,k} \alpha_i \quad \text{and} \quad \kappa_t \stackrel{\text{def}}{=} \max_{\substack{i=0,\dots,t+1 \\ x \in \mathcal{B}_{t+1}}} |q_i(x)|.$$

**Proof.** From (9.5.2), we deduce that

$$\begin{aligned} |m_k(x) - q_k(x)| &= \left| \sum_{i=0}^t (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) q_i(x) \right. \\ &\quad \left. + \sum_{i=t+1}^k (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) q_i(x) - q_k(x) \right|. \end{aligned} \tag{9.5.4}$$

---

<sup>151</sup>“mma” stands for “model with memory approximation”.

We now observe that  $x_{t+1} = x_i = x_k$ , for  $i = t + 1, \dots, k$ , since iteration  $t$  is the last successful iteration. Combining this observation with AM.11, we obtain that

$$q_{t+1}(x) = q_i(x) = q_k(x) \quad (i = t + 1, \dots, k). \quad (9.5.5)$$

Hence,

$$|q_k(x)| \leq \kappa_t. \quad (9.5.6)$$

Substituting (9.5.5) into (9.5.4), we obtain that

$$\begin{aligned} |m_k(x) - q_k(x)| &= \left| \sum_{i=0}^t (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) q_i(x) \right. \\ &\quad \left. + q_k(x) \left[ \sum_{i=t+1}^k (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) - 1 \right] \right| \\ &\leq \kappa_t \sum_{i=0}^t (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) \\ &\quad + \kappa_t \left| \sum_{i=t+1}^k (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) - 1 \right|, \end{aligned} \quad (9.5.7)$$

where we have used the triangle inequality, the definition of  $\kappa_t$ , and (9.5.6). We now consider the last two terms separately. For the first term, we have, from the definition of  $\alpha_{t,k}$  and  $\alpha_i \in [0, 1]$ , that

$$\sum_{i=0}^t (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) \leq \sum_{i=0}^t \left( \prod_{j=i+1}^t \alpha_j \right) \alpha_{t,k}^{k-t} \leq \alpha_{t,k}^{k-t} \sum_{i=0}^t \bar{\alpha}^{t-i} \leq \alpha_{t,k}^{k-t} \sum_{i=0}^{\infty} \bar{\alpha}^i$$

and hence that

$$\sum_{i=0}^t (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) = \frac{\alpha_{t,k}^{k-t}}{1 - \bar{\alpha}}. \quad (9.5.8)$$

This shows that the first term is of the desired order in  $\alpha_{t,k}$ . We now prove, by induction on  $k$  ( $k > t$ ), that this is also the case for the second. To start the induction, we note that

$$\left| \sum_{i=t+1}^{t+1} (1 - \alpha_i) \left( \prod_{j=i+1}^{t+1} \alpha_j \right) - 1 \right| = \alpha_{t+1} \leq \alpha_{t,k}. \quad (9.5.9)$$

Suppose now that

$$\left| \sum_{i=t+1}^{\ell-1} (1 - \alpha_i) \left( \prod_{j=i+1}^{\ell-1} \alpha_j \right) - 1 \right| \leq \alpha_{t,k}^{\ell-1-t} \quad (9.5.10)$$

for  $\ell > t + 1$ . Then

$$\left| \sum_{i=t+1}^{\ell} (1 - \alpha_i) \left( \prod_{j=i+1}^{\ell} \alpha_j \right) - 1 \right|$$

$$\begin{aligned}
&= \left| (1 - \alpha_\ell) + \alpha_\ell \left[ \sum_{i=t+1}^{\ell-1} (1 - \alpha_i) \left( \prod_{j=i+1}^{\ell-1} \alpha_j \right) - 1 + 1 \right] - 1 \right| \\
&= \alpha_\ell \left| \sum_{i=t+1}^{\ell-1} (1 - \alpha_i) \left( \prod_{j=i+1}^{\ell-1} \alpha_j \right) - 1 \right| \\
&\leq \alpha_{t,k}^{\ell-t},
\end{aligned} \tag{9.5.11}$$

where we have used (9.5.10) and the definition of  $\alpha_{t,k}$  to deduce the last inequality. Combining (9.5.9), (9.5.10), and (9.5.11), we see that (9.5.10) is satisfied for all  $\ell \geq t + 1$ , and thus for  $\ell = k + 1$ , yielding

$$\left| \sum_{i=t+1}^k (1 - \alpha_i) \left( \prod_{j=i+1}^k \alpha_j \right) - 1 \right| \leq \alpha_{t,k}^{k-t}.$$

Now substituting this bound and (9.5.8) into (9.5.7), we obtain that

$$|m_k(x) - q_k(x)| \leq \kappa_t \left[ \frac{\alpha_{t,k}^{k-t}}{1 - \bar{\alpha}} + \alpha_{t,k}^{k-t} \right],$$

which gives (9.5.3) with  $\kappa_{\text{mma}} = (2 - \bar{\alpha})/(1 - \bar{\alpha})$ .  $\square$

Thus we see that the model  $m_k$  tends to coincide with the local model  $q_k$  when the number of consecutive unsuccessful iterations increases, the rate of convergence being governed by the value of the memory parameter. This is illustrated in Figure 9.5.1, where the choice  $\alpha_k = 0.4$  was used throughout.

The first picture of this figure (topmost left) shows the contour lines of the model  $m_{k-1}$ , the second (topmost right) the contour lines of the local model  $q_k$ , and the third (second row, left) those of the model  $m_k$ . The subsequent pictures (from left to right and from top to bottom) show the contour lines of  $m_{k+1}, m_{k+2}, m_{k+3}, m_{k+4}$ , and  $m_{k+5}$ , where we have assumed that iterations  $k$  to  $k + 4$  are unsuccessful. The convergence of the models  $m_{k+i}$  to  $q_k$  is clearly visible.

But the memory parameter itself is bounded above by the length of the previous step, as a consequence of AM.11, and thus, indirectly, by the value of the trust-region radius. We exploit this relation in the proof of the following by-now familiar-looking result.

**Theorem 9.5.2** Suppose that AM.10, AM.11, and AN.1 hold. Suppose furthermore that iteration  $k$  is unsuccessful, that the trust-region radius is updated according to (9.1.6), and that the last successful iteration before  $k$  is iteration  $t$ . Then there exist an integer  $\ell(t)$  and a constant  $\kappa_{\text{cnd}}$  such that, for  $k \geq t + \ell(t)$ ,

$$|f(x) - m_k(x)| \leq \kappa_{\text{cnd}} \Delta_k^2 \tag{9.5.12}$$

for each  $x \in \mathcal{B}_k$ .

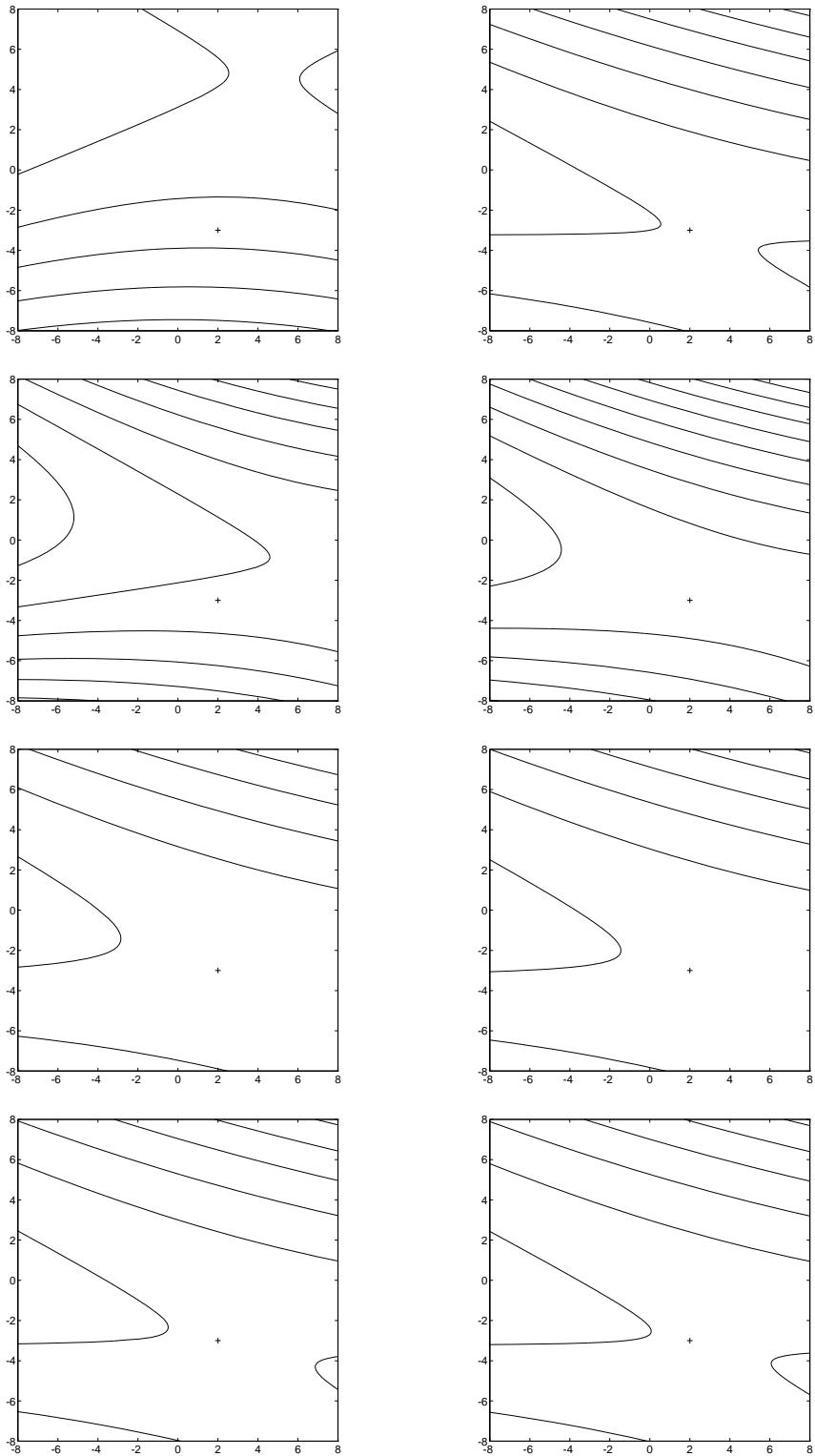


Figure 9.5.1: Contour lines of  $m_{k-1}$ ,  $q_k$ ,  $m_k$ ,  $m_{k+1}$ ,  $m_{k+2}$ ,  $m_{k+3}$ ,  $m_{k+4}$ , and  $m_{k+5}$ , when iterations  $k$  to  $k + 4$  are unsuccessful.

**Proof.** We first note that, because of AM.11,  $q_k$  satisfies the assumptions AM.1–AM.4 made on the model. Hence we may apply Theorem 6.4.1 (p. 133) and deduce that

$$|f(x_k + s_k) - q_k(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_k^2 \quad (9.5.13)$$

for some  $\kappa_{\text{ubh}} > 0$  and for any  $s_k$  such that  $x_k + s_k \in \mathcal{B}_k$ . Now,

$$|f(x) - m_k(x)| \leq |f(x) - q_k(x)| + |q_k(x) - m_k(x)|$$

for all  $x \in \mathcal{B}_k$ . Hence,

$$|f(x) - m_k(x)| \leq \kappa_{\text{ubh}} \Delta_k^2 + \kappa_{\text{mma}} \kappa_t \alpha_{t,k}^{k-t}, \quad (9.5.14)$$

where we have used (9.5.13) and Theorem 9.5.1. Note that  $\alpha_{t,k} \leq \bar{\alpha} < 1$ , and we may therefore define an integer  $\ell_0(t)$  as the smallest positive integer such that  $\kappa_t \bar{\alpha}^{\ell_0} \leq 1$ . As a consequence, (9.5.14) now yields that

$$|f(x) - m_k(x)| \leq \kappa_{\text{ubh}} \Delta_k^2 + \kappa_{\text{mma}} \alpha_{t,k}^{k-t-\ell_0(t)}. \quad (9.5.15)$$

Using AM.10 and AN.1, the inequality  $\|s_{j-1}\| \leq \nu_{j-1}^S \|s_{j-1}\|_{j-1} \leq \kappa_{\text{une}} \Delta_{j-1}$ , and the mechanism of the trust-region radius update, we have that

$$\alpha_j \leq \kappa_\alpha \|s_{j-1}\|^\psi \leq \kappa_\alpha \kappa_{\text{une}}^\psi \Delta_{j-1}^\psi \leq \kappa_\alpha \left( \frac{\kappa_{\text{une}}}{\gamma_1} \right)^\psi \Delta_j^\psi \leq \Delta_j^\psi. \quad (9.5.16)$$

But since all iterations from  $t+1$  to  $k$  are unsuccessful,

$$\Delta_j = \Delta_{t+1}$$

for all  $j = t+1, \dots, k$ . Combining these two observations, we obtain that  $\alpha_{t,k} \leq \Delta_j^\psi$  and therefore

$$\alpha_{t,k}^{k-t-\ell_0(t)} \leq \Delta_j^{\psi(k-t-\ell_0(t))}.$$

Substituting this bound in (9.5.15), we thus deduce that inequality (9.5.12) holds with  $\kappa_{\text{cnd}} = \kappa_{\text{ubh}} + \kappa_{\text{mma}}$  for all  $k$  such that  $k-t \geq \ell_0(t) + \lfloor 2/\psi \rfloor \stackrel{\text{def}}{=} \lfloor \ell(t) \rfloor$ .  $\square$

This shows that the model with memory  $m_k$  is valid in  $\mathcal{B}_k$  after at most  $\ell(t)$  unsuccessful iterations. This indicates that we may use Algorithm 9.1.1 with the very simple feature that model improvement is obtained by simply updating the model by (9.5.1). As a consequence the convergence theory of Section 9.2 applies. There is also an additional simplification in Algorithm 9.1.1. Since the inner iteration at Step 1 is only intended to verify the criticality of the current iterate (by ensuring that  $g_k$  is a reasonable approximation to  $\nabla_x f(x_k)$ ), we may skip this inner iteration altogether since we have assumed that  $\nabla_x f(x_k)$  is available to compute the local model  $q_k$ , which means that first-order criticality of  $x_k$  may be measured directly. However, we should be careful to obtain second-order validity of the model when  $g_k$  is small if we wish to maintain the second-order convergence properties of Algorithm 9.1.1.

There is another case where a more local emphasis on the model would seem to be justified, that is, when the iterates are close to a first-order critical point. In this case, either the critical point is a minimizer and a more local model would then induce a faster rate of convergence for the algorithm, or it is a saddle point or a maximizer, in which case the local negative curvature direction should be found to escape from the neighbourhood quickly. We also note that the condition on  $\kappa_\alpha$  and  $\psi$  in AM.10 is used in (9.5.16) only. Taking these considerations into account, we may consider an alternative condition on the memory parameter, namely, to require that

$$\alpha_k \leq \min[\bar{\alpha}, \Delta_k, \|g_k\|^\theta]$$

for some  $\theta > 0$ . This can be done at the expense of a further slight rearrangement of Algorithm 9.1.1, in that the model  $m_{k+1}$  must then be defined after the trust-region radius update. This does not prevent us from applying the convergence theory developed for conditional models, since  $\mathcal{X}_k$  only contains one vector in our case, and Step 5 of Algorithm 9.1.1 then reduces to choosing  $x_k$  or  $x_k + s_k$  as the next iterate, according to the value of  $\hat{\rho}_k = \rho_k$ . Thus Step 4 can be executed after Step 6 without altering  $x_{k+1}$ , or  $\Delta_{k+1}$ .

The proposal to use models with memory is recent, and numerical experience shows, at this early point, that they are advantageous for some difficult problems in cases where the use of purely local models results in slow progress and a large number of iterations. Of course, this trend needs to be confirmed by further experimentation. As an illustration, we consider the two-dimensional function<sup>152</sup>

$$f(x_1, x_2) = [\sin(\zeta x_1) \sin(\zeta x_2)]^2 + 0.05(x_1^2 + x_2^2), \quad (9.5.17)$$

where  $\zeta$  is a positive parameter. If  $\zeta = 2$ , the average improvement of performance of Algorithm 9.1.1 over Algorithm BTR (p. 116) is about 10%, depending on the starting point. We cannot resist showing the contour lines of this function ( $\zeta = 2$ ) in a neighbourhood of the origin (the solution) in Figure 9.5.2.

In this figure, a trajectory of iterates<sup>153</sup> is also shown, starting from  $(-6, -6.2)$  to the neighbourhood of the solution. In Figures 9.5.3 and 9.5.4, on the following pages, we illustrate the contour lines of the standard “memoryless” Newton models at these points on the left and compare them to those of the models with memory on the right, where we have chosen

$$\alpha_k = \min[0.4, \|x_k - x_{k-1}\|, \|g_k\|^2]. \quad (9.5.18)$$

This is done for all points in the trajectory, starting from the second ( $k = 2$ ), since both models are identical at the first. We note that the two models tend to coincide when the solution is approached (at the last point of the trajectory), since the effect of (9.5.18) is to make  $\alpha_8$  small. We also note that the level curves for the model with memory tend

---

<sup>152</sup>This is problem HUMPS in the CUTE test set. See Bongartz et al. (1995) for a description of this testing environment.

<sup>153</sup>These are not iterates produced by Algorithm BTR, but a sequence of points chosen for illustration.

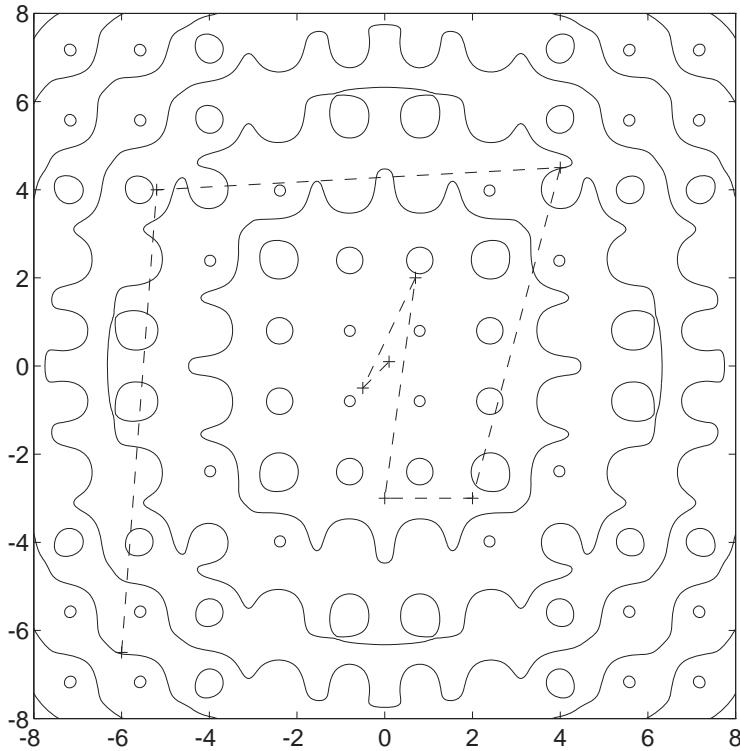


Figure 9.5.2: Contour lines of the function (9.5.17) and the iterates' trajectory.

to appear more “rounded” than for the memoryless models, which is some indication of the beneficial effects of “smoothing” that memory provides. Further computations show that this globally beneficial “smoothing” effect is emphasized as  $\zeta$  increases: the gains in performance are about 50% for  $\zeta = 20$ .

## Notes and References for Section 9.5

The idea of using models with memory was first introduced for both linesearches and trust regions in Gould, Lucidi, Roma, and Toint (1998). Casting it in terms of conditional models is new.

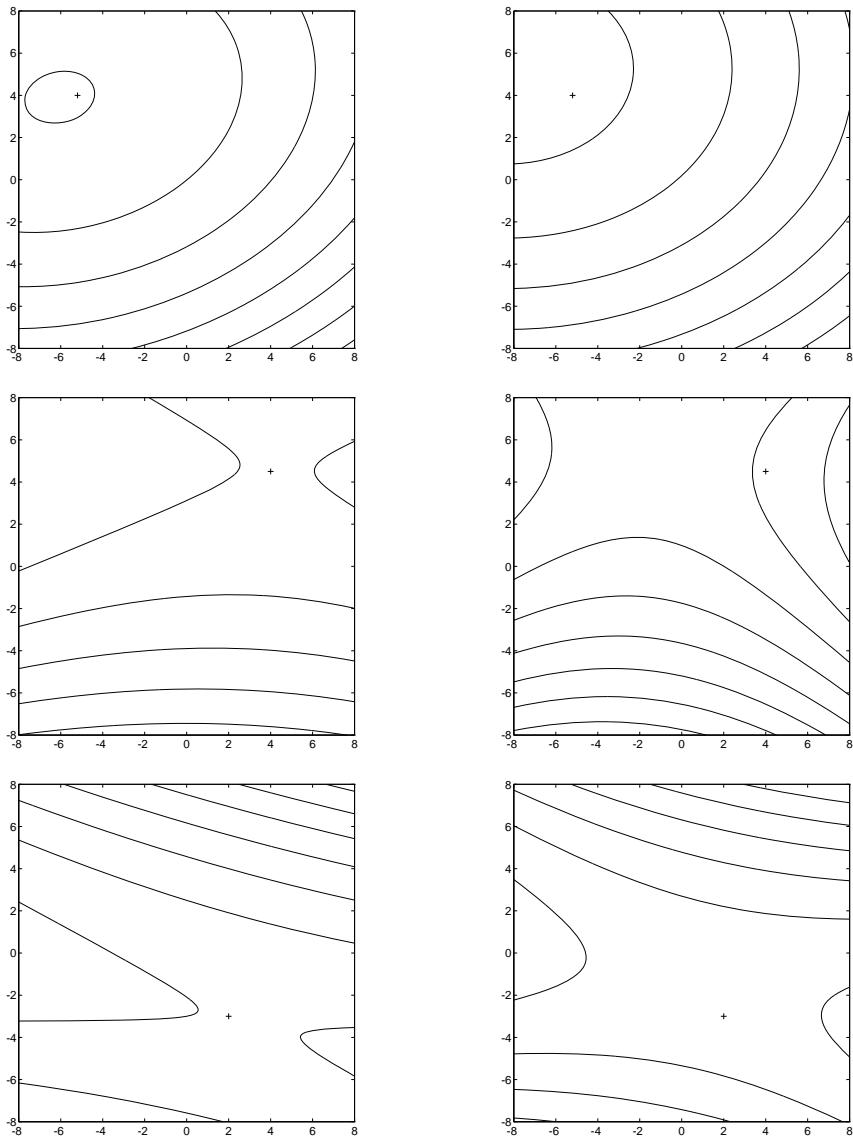


Figure 9.5.3: Contour lines for the memoryless Newton models (on the left) and models with memory (on the right) for points 2 to 4 in the trajectory of Figure 9.5.2.

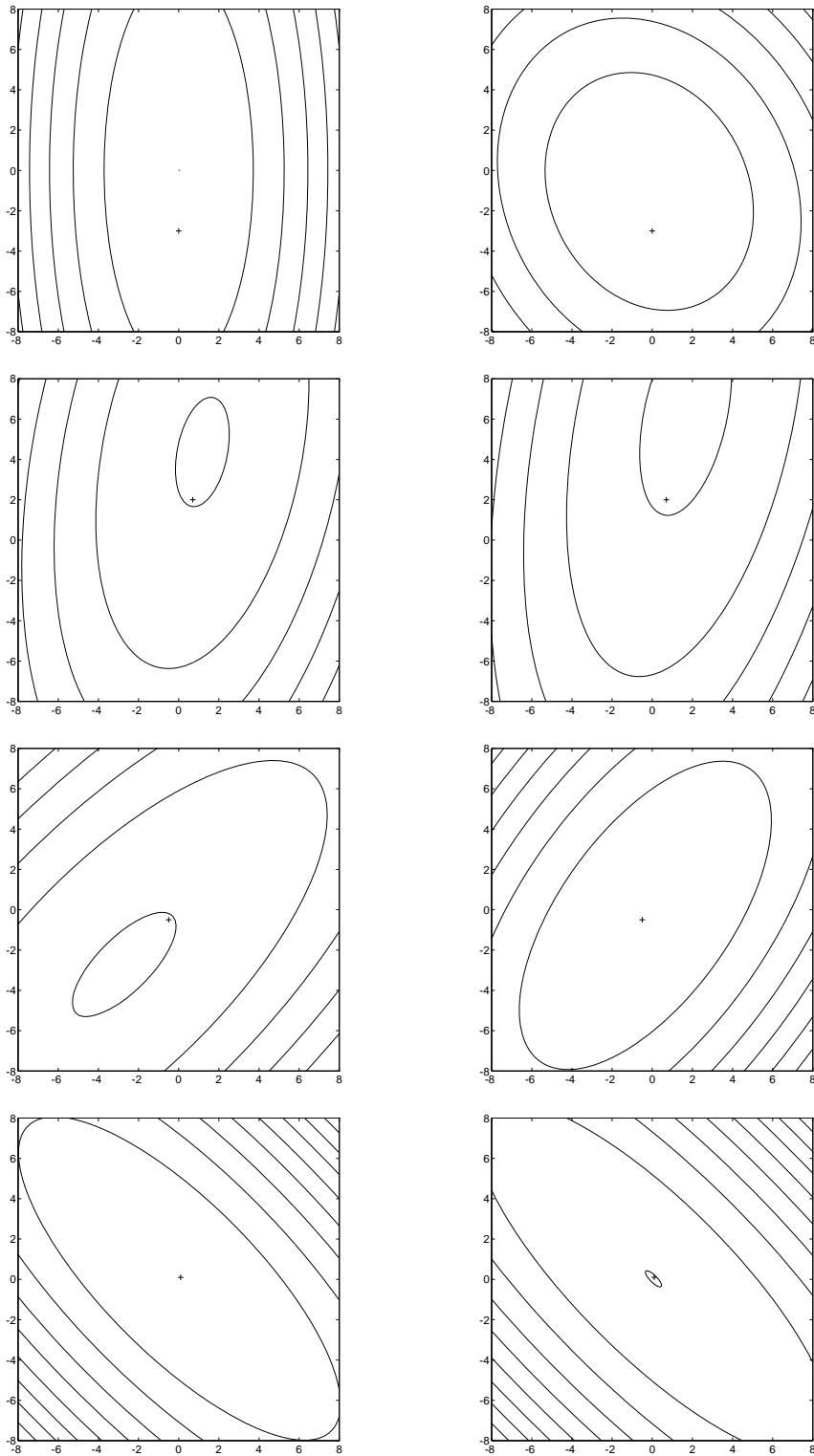


Figure 9.5.4: Contour lines for the memoryless Newton models (on the left) and models with memory (on the right) for points 5 to 8 in the trajectory of Figure 9.5.2.

# Chapter 10

---

## Algorithmic Extensions

---

This chapter is devoted to algorithmic extensions of Algorithm BTR. Although they may be considered as “variants” of this basic algorithm, each of them nevertheless provides improvements that makes them valuable. A busy reader might want to skip this chapter, but we certainly recommend it to the real trust-region devotee.

### 10.1 A Nonmonotone Trust-Region Algorithm

We first consider a variant of our basic trust-region algorithm (Algorithm BTR, p. 116), which turns out to be very useful in practice. All the trust-region algorithms that we have discussed so far are *descent methods*; that is, they only accept the trial point  $x_k + s_k$  as the next iterate if its associated objective function value is strictly lower than that at the current iterate. This monotonicity property ensures that each successful iteration produces a point that is better than any previous point. This property was crucial in our developments of Chapter 6. In this section, we explore the possibility of abandoning this algorithmic restriction. We will thus introduce and analyse a trust-region method for the solution of unconstrained optimization problems such that the sequence  $\{f(x_k)\}$  of function values at the iterates  $x_k$  is no longer monotone. The main advantage is that this will allow the algorithm to be more efficient both because it can “jump over boulders” and because the iterates are not restricted to following the bottom of steep, curved valleys, as often happens when monotonic methods are applied to strongly nonlinear problems. A typical example of this effect is shown in Figure 10.1.1, where the sequences  $\{f(x_k)\}$  are plotted against  $k$  for both a traditional monotone and a nonmonotone trust-region algorithm,<sup>154</sup> applied to a logarithmically

---

<sup>154</sup>The monotone algorithm is the default version of the LANCELOT package of Conn, Gould, and Toint (1992b), and the nonmonotone algorithm is the variant described on p. 357 ( $h = 5$ ), which corresponds to Algorithm NMTR2 in Toint (1997). Both algorithms use a truncated conjugate gradient method to compute the trial step with a banded preconditioner of bandwidth 5. The starting point is  $(0, 1)$ .

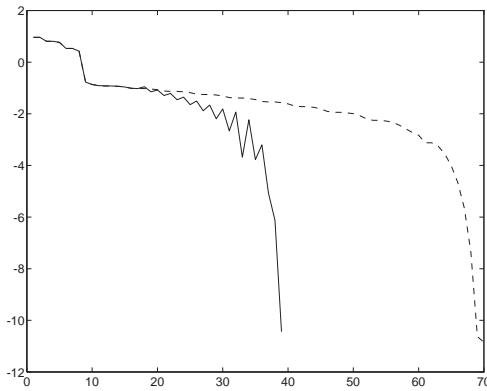


Figure 10.1.1: The value of the objective function on successive iterations for monotone (dashed line) and nonmonotone (continuous line) trust-region algorithms applied on (10.1.1).

rescaled variant of Rosenbrock's function given by

$$f(x_1, x_2) = \log[1 + 10000(x_2 - x_1^2)^2 + (1 - x_1)^2]. \quad (10.1.1)$$

This function has a deep, curved valley following the parabola  $x_2 = x_1^2$ . In fact, the nonmonotone algorithm turns out to be more efficient because it is not constrained to remain at the bottom of this valley, and may instead, to some extent, "climb the sides", allowing it to obtain a better global view of the shape of the objective function.

### 10.1.1 The Nonmonotone Algorithm

The main idea of the nonmonotone trust-region method is to modify Step 3 of Algorithm BTR (p. 116) to allow the choice  $x_{k+1} = x_k + s_k$  even if descent does not occur, that is, if  $\rho_k < 0$ . If we examine the current mechanism in this step, we see that  $f(x_k + s_k) > f(x_k)$  will always prevent acceptance of the trial point, even if the objective function value  $f(x_k + s_k)$  provides a significant improvement relative to other recent iterations. If we wish to give ourselves more freedom, we must modify the definition of the ratio  $\rho_k$  to make the acceptance criterion less myopic. We thus propose not to compare the achieved versus predicted reduction simply during the current iteration, but to capture the trend over some past "history". With this in mind, we define, at iteration  $k$ , a *reference iteration*,<sup>155</sup> which we choose to index by  $r(k) \leq k$ , and restrict our requirement of monotonicity to this subsequence. This means that we only require that, for all  $k$ ,

$$f(x_{r(k)}) < f(x_{r(k-1)}), \quad (10.1.2)$$

where  $r(0) = 0$ . The idea is then to redefine the reference iteration  $r(k)$  from time to time, but with the intent that it should not in principle always be identical to  $k$ .

---

<sup>155</sup>See Section 10.1.3.

The algorithm that we propose to use differs from Algorithm BTR only in that we now consider the objective function decrease from  $x_{r(k)}$  to  $x_k + s_k$  (instead of that from  $x_k$  to  $x_k + s_k$ ) and compare it, using the trust-region philosophy, to some model decrease. For this comparison, we choose to consider the most obvious model decrease we can imagine, that is, the sum of all model decreases corresponding to successful iterations from  $r(k)$  to  $k$  plus the model decrease at iteration  $k$ , to account for the fact that the model changed. The resulting algorithm is Algorithm 10.1.1.<sup>156</sup>

**Algorithm 10.1.1: Nonmonotone trust-region algorithm**

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given as in Algorithm BTR (p. 116). Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m_k$  in  $\mathcal{B}_k$ .

**Step 2: Step calculation.** Compute a step  $s_k$  that sufficiently reduces the model  $m_k$  in the sense of AA.1 and such that  $x_k + s_k \in \mathcal{B}_k$ .

**Step 3: Acceptance of the trial point.** Define the reference iteration  $r(k) \leq k$  such that (10.1.2) holds, and compute  $f(x_k + s_k)$ , the cumulative model decrease

$$\sigma_k^h = \sum_{\substack{i=r(k) \\ i \in \mathcal{S}}}^{k-1} [m_i(x_i) - m_i(x_i + s_i)],$$

and the ratios

$$\rho_k^h = \frac{f(x_{r(k)}) - f(x_k + s_k)}{\sigma_k^h + m_k(x_k) - m_k(x_k + s_k)} \text{ and } \rho_k^c = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Set

$$\rho_k = \max[\rho_k^h, \rho_k^c]. \quad (10.1.3)$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k) & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

The superscripts “h” and “c” in  $\sigma_k^h$ ,  $\rho_k^h$ , and  $\rho_k^c$  are meant to be mnemonics for “historical” and “current”. Note that the value of  $\rho_k$  given by (10.1.3) is never smaller

---

<sup>156</sup>For the sake of simplicity in the statement of this algorithm, we have ignored the formal possibility that  $m_k(x_k) = m_k(x_k + s_k)$ , which only arises if  $x_k$  is already first- or second-order critical, as can be seen from Theorems 6.3.4 (p. 131) and 6.6.3 (p. 154).

than that calculated by Algorithm BTR ( $\rho_k^c$ ). Hence the effect of the new Step 3 is to accept steps that would be rejected by the original algorithm, provided they still guarantee a sufficient decrease of the objective function over *some* past set of iterations. The purpose of such steps is simply that they might improve the overall efficiency of the minimization, even if they appear unappealing from a local perspective. Algorithm 10.1.1 is similar in spirit to the “watchdog technique” (see the notes at the end of this section [p. 358] and at the end of Section 15.3.2 [p. 654]), where the idea is to let the algorithm run free for a few iterations, to evaluate the result, and to correct for erratic behaviour by backtracking to a previous reference iterate only as a last resort.

For a practical algorithm, it is of course essential to specify how the reference iteration is chosen at Step 3. We will discuss possible choices later, but we first consider the general framework and see how the convergence results of Chapter 6 can be adapted to handle the nonmonotone algorithm.

### 10.1.2 Convergence Theory Revisited

We first revisit the notion of a successful iteration and state the following simple but crucial property.

**Lemma 10.1.1** Suppose that AA.1 holds. Then, for each  $k \in \mathcal{S}$ ,

$$f(x_0) - f(x_{k+1}) \geq \eta_1 \kappa_{\text{mdc}} \sum_{\substack{t=0 \\ t \in \mathcal{S}}}^k \|g_t\| \min \left[ \frac{\|g_t\|}{\beta_t}, \Delta_t \right]. \quad (10.1.4)$$

**Proof.** We immediately deduce, from AA.1, the definition (10.1.3), the concept of successful iteration, and the fact that  $\sigma_k^h > 0$ , that

$$f(x_{p(k)}) - f(x_{k+1}) \geq \eta_1 \kappa_{\text{mdc}} \sum_{\substack{j=p(k) \\ j \in \mathcal{S}}}^k \|g_j\| \min \left[ \frac{\|g_j\|}{\beta_j}, \Delta_j \right], \quad (10.1.5)$$

where

$$p(k) = \begin{cases} r(k) & \text{when } \rho_k^h \geq \rho_k^c, \\ k & \text{otherwise.} \end{cases}$$

Now consider any iteration of index  $k \in \mathcal{S}$ . We see that this iteration has an associated reference iteration  $p(k)$  whose iterate  $x_{p(k)}$  has been obtained by a previous successful iteration, which in turn has an associated reference iteration, and so on, back to the stage where  $x_0$  is reached by this backwards referencing process. Hence we may construct, for each  $k \in \mathcal{S}$ , a sequence of successful iterates indexed by  $\{p_1, p_2, \dots, p_q\}$  such that

$$x_0 = x_{p(p_1)} \text{ and } x_{p_{j-1}+1} = x_{p(p_j)} \text{ for } j = 2, \dots, q \text{ and } x_{p_q} = x_{p(k)}. \quad (10.1.6)$$

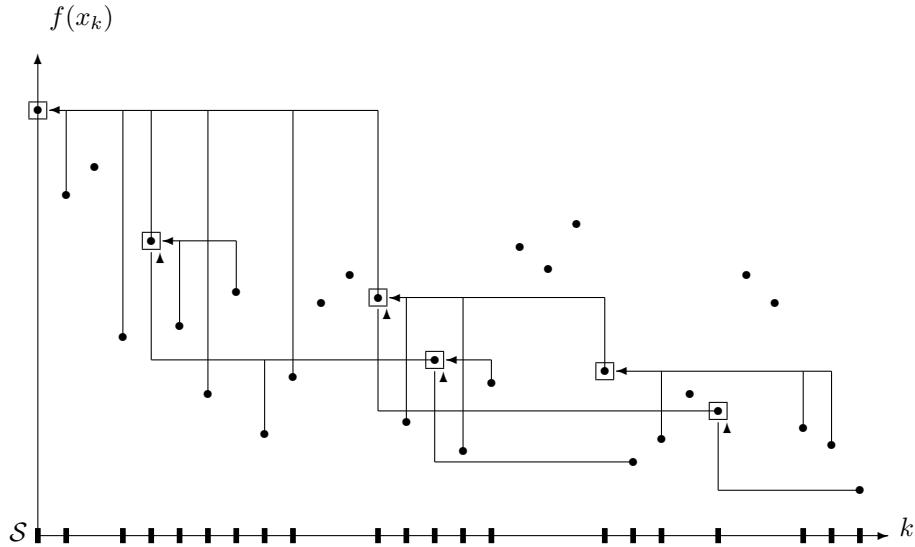


Figure 10.1.2: The backward referencing subsequences.

The backwards referencing process is illustrated in Figure 10.1.2, where the values of the objective function at reference iterations are shown in small boxes. Notice the possible interleaving of the referencing subsequences.

Now observe that

$$f(x_0) - f(x_{k+1}) = f(x_0) - f(x_{p_1+1}) + \sum_{j=2}^q [f(x_{p_{j-1}+1}) - f(x_{p_j+1})] + f(x_{p(k)}) - f(x_{k+1})$$

and (10.1.5) implies that not only is each difference on the right-hand side of this equation positive, but also

$$f(x_0) - f(x_{p_1+1}) \geq \kappa_{\text{mdc}} \eta_1 \sum_{\substack{t=0 \\ t \in \mathcal{S}}}^{p_1} \|g_t\| \min \left[ \frac{\|g_t\|}{\beta_t}, \Delta_t \right], \quad (10.1.7)$$

$$f(x_{p_{j-1}+1}) - f(x_{p_j+1}) \geq \kappa_{\text{mdc}} \eta_1 \sum_{\substack{t=p_{j-1}+1 \\ t \in \mathcal{S}}}^{p_j} \|g_t\| \min \left[ \frac{\|g_t\|}{\beta_t}, \Delta_t \right]$$

for  $j = 2, \dots, q$ , and

$$f(x_{p(k)}) - f(x_{k+1}) \geq \kappa_{\text{mdc}} \eta_1 \sum_{\substack{t=p(k) \\ t \in \mathcal{S}}}^k \|g_t\| \min \left[ \frac{\|g_t\|}{\beta_t}, \Delta_t \right]. \quad (10.1.8)$$

Summing (10.1.7)–(10.1.8), we obtain (10.1.4).  $\square$

We now wish to show that Theorem 6.4.2 (p. 134) still holds for Algorithm 10.1.1, and thus that progress can be made if the current iterate is not first-order critical.

**Theorem 10.1.2** Suppose that AF.1, AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Suppose furthermore that  $g_k \neq 0$  and that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\| (1 - \eta_2)}{\kappa_{\text{ubh}}}.$$

Then iteration  $k$  is very successful and

$$\Delta_{k+1} \geq \Delta_k.$$

**Proof.** This can be derived simply from the proof of Theorem 6.4.2 (p. 134), where inequality (6.4.12) gives us that  $\rho_k^c \geq \eta_2$ . The definition (10.1.3) then ensures that  $\rho_k \geq \eta_2$  and the conclusion follows.  $\square$

As a consequence of this property, we may now pursue our convergence theory exactly as in Section 6.4, at least up to the proof of Theorem 6.4.5 (p. 136), where equation (6.4.17) is no longer valid due to the different definition of a successful iteration for Algorithm 10.1.1. We therefore reexamine this result, although the proof is actually very similar to the original.

**Theorem 10.1.3** Suppose that AF.1–AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Then one has that

$$\liminf_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0. \quad (10.1.9)$$

**Proof.** First note that Theorem 6.4.4 (p. 136) yields the desired conclusion if the number of successful iterations is finite. Suppose thus that this is not the case, and also, for the purpose of obtaining a contradiction, that

$$\|\nabla_x f(x_k)\| = \|g_k\| \geq \epsilon, \quad (10.1.10)$$

for all  $k$  and for some  $\epsilon > 0$  independent of  $k$ . Then consider a successful iteration with index  $k$ . From condition (10.1.4), the fact that  $\beta_k$  is bounded above by AM.4, and using (6.3.2) (p. 124), (10.1.10), and Theorem 6.4.3 (p. 135), we have that

$$f(x_0) - f(x_{k+1}) \geq \sigma_k \eta_1 \kappa_{\text{mdc}} \epsilon \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}} \right] > 0, \quad (10.1.11)$$

where

$$\sigma_k \stackrel{\text{def}}{=} |\mathcal{S} \cap \{1, \dots, k\}|$$

is the number of successful iterations up to and including iteration  $k$ . Since the objective function is bounded below, by AF.2, the left-hand side of (10.1.11) is bounded by a quantity independent of  $k$ , and  $\sigma_k$  must therefore satisfy

$$\sigma_k \leq \bar{\sigma}$$

for some  $\bar{\sigma}$  independent of  $k$ . But this contradicts the assumption that there are infinitely many successful iterations. Thus (10.1.10) cannot hold for all  $k$  and (10.1.9) follows.  $\square$

We note that Theorem 6.4.4 (p. 136) still applies without modification. We next prove that, as before, all limit points are first-order critical.

**Theorem 10.1.4** Suppose that AF.1–AF.3, AN.1, AM.1–AM.4, and AA.1 hold. Then one has that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

**Proof.** Assume, again for the purpose of obtaining a contradiction, that there is a subsequence of successful iterates, indexed by  $\mathcal{N}$ , such that

$$\|\nabla_x f(x_k)\| = \|g_k\| \geq 2\epsilon > 0 \quad (10.1.12)$$

for some  $\epsilon > 0$  and for all  $k \in \mathcal{N}$ , and define  $\mathcal{K}$  to be the index set of all successful iterations such that  $\|g_k\| \geq \epsilon$ . Note that  $\mathcal{N} \subseteq \mathcal{K}$ . Consider a  $k \in \mathcal{K}$ . Applying (10.1.4), the inclusion  $\mathcal{K} \subseteq \mathcal{S}$ , and the definition of  $\mathcal{K}$ , we obtain that

$$\begin{aligned} f(x_0) - f(x_{k+1}) &\geq \kappa_{\text{mdc}} \eta_1 \sum_{\substack{t=0 \\ t \in \mathcal{S}}}^k \|g_t\| \min \left[ \frac{\|g_t\|}{\kappa_{\text{umh}}}, \Delta_t \right] \\ &\geq \kappa_{\text{mdc}} \eta_1 \sum_{\substack{t=0 \\ t \in \mathcal{K}}}^k \|g_t\| \min \left[ \frac{\|g_t\|}{\kappa_{\text{umh}}}, \Delta_t \right] \\ &\geq \kappa_{\text{mdc}} \eta_1 \epsilon \sum_{\substack{t=0 \\ t \in \mathcal{K}}}^k \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_t \right]. \end{aligned} \quad (10.1.13)$$

But the left-hand side of this inequality is bounded because  $f(x)$  is bounded below, by AF.2, and we see thus that

$$\min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_t \right] = \frac{\epsilon}{\kappa_{\text{umh}}}$$

can only be true a finite number of times. Thus there must exist a  $k_0 \in \mathcal{K}$  such that

$$\min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_t \right] = \Delta_t$$

for all  $t \in \mathcal{K}$ ,  $t \geq k_0$ . Therefore, (10.1.13) implies that

$$\sum_{\substack{t=k_0 \\ t \in \mathcal{K}}}^k \Delta_t \leq \frac{1}{\kappa_{\text{mdc}} \eta_1 \epsilon} \left( f(x_0) - \min_{x \in \mathbb{R}^n} f(x) \right)$$

for all  $k \in \mathcal{K}$ ,  $k \geq k_0$ . As a consequence,

$$\sum_{\substack{t=k_0 \\ t \in \mathcal{K}}}^{\infty} \Delta_t < \infty. \quad (10.1.14)$$

Theorem 10.1.3 and the definition of  $\mathcal{K}$  now imply that, for any  $k \in \mathcal{N} \subseteq \mathcal{K}$ , there must be a smallest  $\ell(k) > k$  with  $\ell(k) \in \mathcal{S} \setminus \mathcal{K}$ . Furthermore, since

$$\|x_{\ell(k)} - x_k\| \leq \sum_{\substack{j=k \\ j \in \mathcal{S}}}^{\ell(k)-1} \|x_{j+1} - x_j\| \leq \sum_{\substack{j=k \\ j \in \mathcal{S}}}^{\ell(k)-1} \nu_j^s \Delta_j = \sum_{\substack{j=k \\ j \in \mathcal{K}}}^{\ell(k)-1} \nu_j^s \Delta_j \leq \kappa_{\text{une}} \sum_{\substack{j=k \\ j \in \mathcal{K}}}^{\ell(k)-1} \Delta_j,$$

by AN.1, we obtain from (10.1.14) that  $\|x_{\ell(k)} - x_k\|$  tends to zero as  $k$  tends to infinity. By continuity of the gradient, AF.1, and AM.3, we thus deduce that  $\|g_k\| - \|g_{\ell(k)}\|$  also tends to zero with  $k$ . However, this is impossible because the definitions of  $\mathcal{N}$ ,  $k$ , and  $\ell(k)$  imply that  $\|g_k\| - \|g_{\ell(k)}\| \geq \epsilon$ . Hence, no subsequence satisfying (10.1.12) can exist, and the theorem is proved.  $\square$

The global convergence of Algorithm 10.1.1 to first-order critical points is thus ensured. We may now examine its convergence to second-order critical points. A careful reading of Section 6.5 shows that all results in this section remain valid, provided we make the necessary step of replacing  $\rho_k$  by  $\rho_k^c$  in the proof of Lemma 6.5.3 (p. 145). Hence we also conclude that the complete sequence converges to an isolated minimizer if there is one among the limit points, and that the trust-region radius ultimately becomes irrelevant, allowing for a fast rate of convergence.

The situation is similar if we consider using nonconvex models. The content of Section 6.6 still applies to Algorithm 10.1.1 without modification, up to Theorem 6.6.4 (p. 154). The proof of this theorem is easily generalized by replacing (6.6.23) (p. 155) by

$$f(x_0) - f(x_{k_0+j+1}) \geq \frac{1}{2} \sigma_{k_0, k_0+j} \eta_1 \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_2^2] > 0,$$

where

$$\sigma_{t,k} \stackrel{\text{def}}{=} |\mathcal{S} \cap \{t, \dots, k\}| \quad (t \leq k)$$

is the number of successful iterations indexed between  $t$  and  $k$ . Lemma 6.6.6 (p. 156) and its proof also remain valid if we remember, when deriving (6.6.29) (p. 157), that  $\sigma_k$  is always nonnegative in Algorithm 10.1.1. We then obtain Theorem 6.6.7 (p. 157) without further modifications. Theorem 6.6.8 (p. 159) may also be extended to cover Algorithm 10.1.1 if we again replace  $\rho_k$  by  $\rho_k^c$  when we are analysing the successful iteration  $k$ , and rephrase (6.6.41), (6.6.44), (6.6.46) (p. 160), and (6.6.48) (p. 161) as

$$f(x_0) - f(x_{k+j+1}) \geq \frac{1}{2} \sigma_{\ell, \ell+j} \eta_1 \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_2^2] > 0,$$

$$f(x_0) - f(x_{q+1}) \geq f(x_j) - f(x_{j+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_4^2] \stackrel{\text{def}}{=} \delta_3 > 0,$$

$$f(x_0) - f(x_{q+1}) \geq f(x_{j-1}) - f(x_j) \geq \eta_1 \frac{1}{2} \kappa_{\text{sod}} |\lambda_*| \min \left[ \frac{1}{4} \lambda_*^2, \left( \frac{\delta_4}{\gamma_4} \right)^2 \right] \stackrel{\text{def}}{=} \delta_5 > 0,$$

and

$$f(x_0) - f(x_{q+1}) \geq f(x_q) - f(x_{q+1}) \geq \eta_1 \frac{1}{2} \kappa_{\text{sol}} |\lambda_*| \min[\frac{1}{4} \lambda_*^2, \delta_6^2] \stackrel{\text{def}}{=} \delta_7 > 0,$$

respectively. In the first case, the argument is essentially unchanged in that one sees that  $\sigma_{\ell, \ell+j}$  cannot tend to infinity because of AF.2. In the last three cases one then needs to note that  $f(x_0) - f(x_{q+1})$  is increased by  $\min[\delta_3, \delta_5, \delta_7]$  every time the sequence of iterates leaves  $\mathcal{B}(x_*, \delta)$ . This cannot happen infinitely often because of AF.2, and we then deduce a contradiction as in Theorem 6.6.8 (p. 159).

We have thus shown that the modification of Step 3 defining Algorithm 10.1.1 is consistent with the general framework of trust-region methods, in that *all standard convergence properties are preserved*.

### 10.1.3 The Reference Iteration and Other Practicalities

Having established the theoretical coherence of our proposal for a nonmonotone trust-region algorithm, we now discuss some possibilities for the practical choice of a reference iteration  $r(k)$ , which we require at the beginning of Step 3 in Algorithm 10.1.1. Such techniques for choosing the reference iteration may be viewed as heuristics, in the sense that reliability and efficiency of the overall algorithm must still be evaluated by experimentation, but the theoretical results of course still hold. We now present some proposals and discuss their relative merits in view of numerical experience to date.

The simplest way to select the reference iteration is to look back in history using a “sliding window”, considering, say, the last  $m$  iterations. Over these iterations, it is natural to select that which gives rise to the highest objective function value, so as to give the algorithm the best chance of accepting  $x_k + s_k$  as the next iterate. In other words, we select  $r(k)$  from the successful iterations by requiring that

$$f(x_{r(k)}) = \max_{i=0, \dots, \min[k, m]} f(x_{k-i}).$$

This rule is easy to implement in a practical algorithm, although it requires the storage of as many as  $m$  function values and model decreases—as  $m$  is typically rather low, this additional storage is acceptable. Similarly, the overhead in computing Step 3 of Algorithm 10.1.1 compared to Step 3 of Algorithm BTR (p. 116) is of the order of  $m$  operations, which may be considered negligible compared to the cost of computing the step  $s_k$ , except possibly for very small problems.

This technique appears to be reasonably effective for values of  $m$  around 25, but experiments indicate that difficulties may arise when the objective function values increase over several successive iterations having reached a (temporary) minimum, and that one might prefer the following strategy. Let  $f_{\min}$  denote the current best overall value of the objective function, that is, at iteration  $k$ ,

$$f_{\min} = \min_{i=0, \dots, k} f(x_i).$$

Also let there have been  $\ell$  successful iterations since the value of  $f_{\min}$  was first attained. We then reset the reference iteration and its associated function value, which we denote

$f_r$  if  $\ell$  exceeds some preset positive integer constant  $m$ . When this occurs, we suggest resetting  $f_r$  to the largest value of the objective function observed over all successful iterations, including the current one, starting from the one iteration at which  $\ell$  was reset to zero (i.e., since the last overall best value was found). We denote this maximal value by  $f_c$ . The resulting backward referencing subsequences are shown in Figure 10.1.3 for  $m = 2$ , where, as in Figure 10.1.2, the objective values at reference iterations are boxed and where we have circled the successive values of  $f_{\min}$ . We also report, in this figure, the value of the index  $\ell$  at each iteration and indicate the iterations at which the reference value is reset by a star.

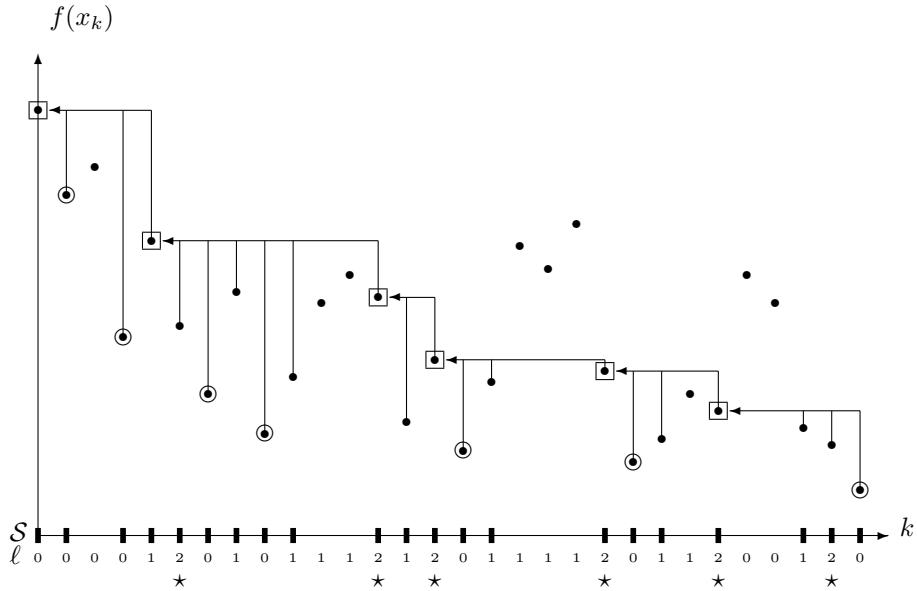


Figure 10.1.3: The backward referencing subsequences when the reference iteration is chosen as in Algorithm 10.1.2.

We must thus remember the candidate reference value  $f_c$  and the associated sum of predicted model decreases on all subsequent successful iterations (denoted  $\sigma_c$ ). We finally denote by  $\sigma_r$  the sum of all predicted model decreases on the successful iterations since the reference iteration. If we suppose, in the formal description that follows, that the initializations

$$f_{\min} = f_r = f_c = f(x_0), \quad \ell = 0, \quad \text{and } \sigma_r = \sigma_c = 0$$

are performed at Step 0 of the algorithm, we may specify our strategy as Algorithm 10.1.2.<sup>157</sup>

Note that when  $f_r$  is reset at Step 3c, at least one of  $\rho_k^h$  or  $\rho_k^c$  must be positive. In the first case this immediately implies that  $f(x_{k+1}) = f(x_k + s_k) < f_r$ , and (10.1.2)

---

<sup>157</sup>We again ignore the formal but uninteresting possibility that  $m_k(x_k) = m_k(x_k + s_k)$ .

**Algorithm 10.1.2: Step 3 of a nonmonotone trust-region algorithm****Step 3: Acceptance of the trial point.**

**Step 3a: Update the iterate.** Compute  $f(x_k + s_k)$  and the ratios

$$\rho_k^h = \frac{f_r - f(x_k + s_k)}{\sigma_r + m_k(x_k) - m_k(x_k + s_k)} \text{ and } \rho_k^c = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

and set

$$\rho_k = \max[\rho_k^h, \rho_k^c].$$

If  $\rho_k < \eta_1$ , then define  $x_{k+1} = x_k$  and go to Step 4. Otherwise define  $x_{k+1} = x_k + s_k$  and update  $\sigma_c$  and  $\sigma_r$  according to

$$\sigma_c = \sigma_c + m_k(x_k) - m_k(x_{k+1}) \text{ and } \sigma_r = \sigma_r + m_k(x_k) - m_k(x_{k+1}).$$

**Step 3b: Update the best value found so far.**

If  $f(x_{k+1}) < f_{\min}$  then set

$$f_c = f_{\min} = f(x_{k+1}), \quad \sigma_c = 0, \quad \text{and} \quad \ell = 0,$$

and go to Step 4. Otherwise, increment  $\ell$  by 1.

**Step 3c: Update the candidate for the next reference value.**

If  $f(x_{k+1}) > f_c$ , set

$$f_c = f(x_{k+1}) \text{ and } \sigma_c = 0.$$

**Step 3d: Possibly reset the reference value.**

If  $\ell = m$ , set

$$f_r = f_c \text{ and } \sigma_r = \sigma_c.$$

therefore holds. In the second case,  $f(x_{k+1}) = f(x_k + s_k) < f(x_k)$ . But the same reasoning may be applied to the successful iteration at which  $x_k$  was accepted and to all successful iterations since  $f_r$  was reset for the last time. Thus we deduce that  $f(x_k) < f_r$ , and (10.1.2) always holds.

Observe that Algorithm 10.1.2 is a special case of Step 3 of Algorithm 10.1.1 (with the slight change that  $\sigma_k$  is no longer indexed by iteration number, but referred to as  $\sigma_r$  in order to emphasize that it can be recurred from iteration to iteration). As a consequence, the theoretical properties of Algorithm 10.1.1 are also obtained for this variant. It is enough to mention that Toint (1997) and Xiao (1996) indicate that the use of nonmonotone techniques such as those discussed here may represent an average improvement, over monotone forms of other state-of-the-art trust-region algorithms, of approximately 30% in iterations and objective function evaluations, and approximately 15% in cpu time, while maintaining their excellent reliability.

There are a number of additional useful algorithmic extensions that are possible in this context. For instance, we could alter the reference value for the objective function at the first few iterations to allow an initial *increase* of the objective. This sometimes produces substantial gains in efficiency, as is shown by Figure 10.1.4.<sup>158</sup> Xiao (1996) also suggests detailed strategies to vary the history length  $m$  from iteration to iteration, depending on the number of successive iterations for which the best function value remains unimproved, or to force monotonicity when either a significant decrease has just been obtained or if an objective function decrease appears to stagnate. Despite the initially favourable tests, the true value of these heuristics remains to be verified by continued numerical experimentation, but one can safely conclude at this point that nonmonotone methods do often provide a significant improvement in practice over their monotone counterparts.

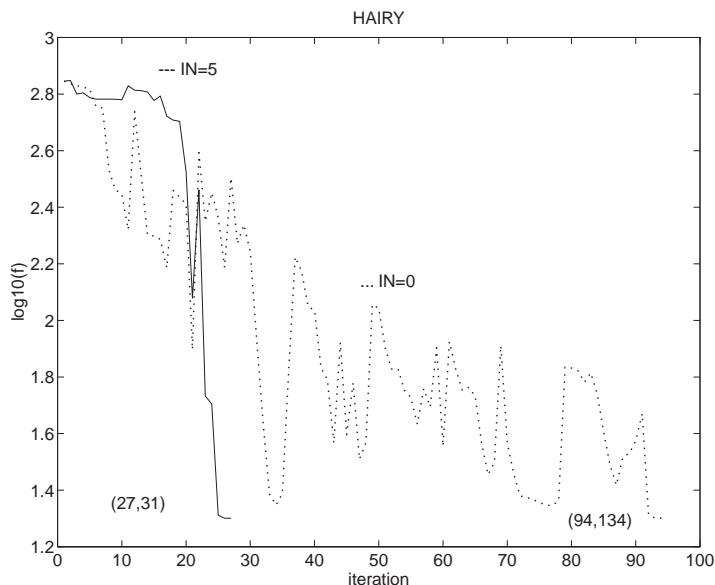


Figure 10.1.4: The effect of allowing a reference value for the objective function larger than  $f(x_0)$  in early iterations, for problem HAIRY of the CUTE<sup>159</sup> test set (IN=5), compared to insisting on descent with respect to  $f(x_0)$  (IN=0). Here, IN is the number of iterations at the beginning of the algorithm, where no restriction is put on the value of the objective function.

## Notes and References for Section 10.1

Perhaps the first nonmonotone optimization technique is the “watchdog technique” introduced by Chamberlain et al. (1982) in order to avoid the so-called Maratos effect in certain

---

<sup>158</sup>This figure is extracted from Xiao (1996), p. 58, courtesy of the author.

<sup>159</sup>See Bongartz et al. (1995).

constrained optimization methods (see the notes at the end of Section 15.3.2). This idea was developed further in a sequence of seminal papers by Grippo, Lampariello, and Lucidi (1986, 1989, 1991), in which linesearch methods for unconstrained problems were investigated. Their conclusions were recently revisited by Toint (1996). Other references concerning non-monotone linesearch techniques include Panier and Tits (1991), Bonnans et al. (1992), and Facchinei and Lucidi (1993). The work reported on in Section 10.1 is based on Toint (1994, 1997), which resulted from the natural question of extending the idea from linesearch algorithms to trust-region methods. This idea had also been explored by Xiao and Zhou (1992), Deng, Xiao, and Zhou (1993), Zhou and Xiao (1994), and Xiao and Chu (1995), although from the point of view of adapting nonmonotone linesearch conditions for their use in the trust-region framework. Further contributions on nonmonotone trust-region algorithms can be found in Ke and Han (1995a, 1996), Ke, Liu, and Xu (1996), Burke and Weigmann (1997), and Yuan (1999). The subject is also mentioned by Hanson and Krogh (1992) in the context of constrained nonlinear least-squares problems. Examples of nonmonotone trust-region algorithms for generally constrained problems include the filter method described in Section 15.5 and the proposal by Ulbrich and Ulbrich (1999).

## 10.2 Structured Trust Regions

### 10.2.1 Motivation and Basic Concepts

So far, all the algorithms we have considered have used a *single* trust-region radius to control the models employed, even if the objective function is a composite function of several different components. This decision is somewhat surprising if one accepts that some of the components modelled might be substantially “better behaved” than others, since this implies that the region in which their corresponding models ought to be trusted should also be significantly larger. In this case, the choice of a single (unstructured) trust region may be viewed as a conservative strategy ensuring that *all* models are trustworthy in what amounts to a “safe minimal” region. While this strategy might be reasonable for small problems, where each of the functions involved usually depends on all or most of the problem’s variables, it is more questionable for large-scale applications, where an individual component function typically depends on only a small number of variables. For instance, one might consider the minimization of an unconstrained objective function consisting of the sum of many quadratic and a few highly nonlinear terms, the latter involving a small subset of the variables. If a classical unstructured trust-region algorithm with a quadratic model is used, the quadratic terms may be perfectly modelled, but the steps one can make at each iteration are (unnecessarily) limited by the highly nonlinear behaviour in a small subset of the variables.

A second but related reason for taking the structure of the problem into account is apparent when considering the *separable* unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x) = \sum_{i=1}^p f_i(x_i), \quad (10.2.1)$$

where the vectors  $x_i \in \mathbb{R}^{n_i}$  and  $\mathbb{R}^n = \prod_{i=1}^p \mathbb{R}^{n_i}$ . In this case, we actually have  $p$  completely independent unconstrained minimization problems, which could (and ideally should) be solved separately. If we use Algorithm BTR (p. 116) for each of the  $p$  minimizations, the decisions to contract or expand the trust-region radii for each of these minimizations are obviously independent for  $i = 1, \dots, p$  and only related to the behaviour of the model of each  $f_i$ . If, for some reason, one prefers to minimize  $f$  as a single composite function, it certainly makes sense to try to preserve this independence as far as possible.

It is the purpose of this section to present and analyse a trust-region algorithm that uses the *structure* of the problem in the definition of the trust region. This algorithm, to a large extent, retains the independence discussed in the previous paragraph and certainly allows large steps in directions in which the model has proved to be adequate while restricting the movement in directions where the model seems unreliable. To be more precise, we will consider the problem of minimizing a *partially separable* objective function.

We say that  $f$  is partially separable whenever

$$f(x) = \sum_{i=1}^p f_i(x) \quad (10.2.2)$$

and, for each  $i \in \{1, \dots, p\}$ , there exists an *invariant* subspace  $\mathcal{N}_i \neq \{0\}$  such that, for all  $w \in \mathcal{N}_i$  and all  $x \in \mathbb{R}^n$ ,

$$f_i(x + w) = f_i(x).$$

If this is the case, the  $f_i$  are called the *element functions*, or *elements*, of  $f$  and the *range subspaces* associated with each element function  $f_i$  are defined as

$$\mathcal{R}_i \stackrel{\text{def}}{=} \mathcal{N}_i^\perp,$$

where  $\mathcal{N}_i^\perp$  denotes the subspace of vectors orthogonal to  $\mathcal{N}_i$ . We are mostly interested in the case where the dimension of each  $\mathcal{R}_i$  is small compared to  $n$ . A commonly occurring case is when each element function  $f_i$  only depends on a small subset of the problem's variables:  $\mathcal{R}_i$  is then the subspace spanned by the vectors of the canonical basis corresponding to the variables that occur in  $f_i$  (the *elemental variables*). The range of the projection operator  $P_{\mathcal{R}_i}(\cdot)$  is therefore of low dimensionality. We note that  $f$  is invariant for any translation in the subspace  $(\sum_{i=1}^p \mathcal{R}_i)^\perp$ . We may therefore restrict our attention to the case where  $\sum_{i=1}^p \mathcal{R}_i = \mathbb{R}^n$  without loss of generality.

We use the decomposition of the objective function into element functions as the basis for our definition of a structured trust region. The reason for concentrating on partially separable structures is that partial separability is in fact a very general geometric<sup>160</sup> structure, occurring quite naturally in a large number of different applications. More significantly, partial separability provides a decomposition<sup>161</sup> of a given

<sup>160</sup>In the sense that it does not depend on the choice of a basis.

<sup>161</sup>It can be shown that all sufficiently smooth functions with a sparse Hessian matrix are partially separable, and also that partially separable functions are dense in the set of twice-continuously differ-

nonlinear function into a linear combination of smaller element functions, each of which may then be modelled separately. It is thus quite natural to assign one trust-region radius per element function, and to decide upon its increase or decrease separately. Because different element functions typically involve different sets of variables, each *element trust region* only restricts the components of the step corresponding to its elemental variables.

An obvious alternative approach that takes structure into account is to use a scaling matrix, just as we did in Section 6.7, to account for differences in the quality of the models of elements when constructing the overall trust region. This would be satisfactory if our previous theory did not require, in effect, that the sequence of scaling matrices have a uniformly bounded condition number (see Theorem 6.7.1). Unfortunately, it is easy to conceive of instances where this might be a severe handicap. For example, it would prevent the trust region in the subspace of variables corresponding to well-modelled (perhaps linear or quadratic) elements from continuing to expand while ensuring that the region in the subspace of variables from badly behaved nonlinear element functions remains modest. Furthermore, this strategy may well lead to numerical difficulties when attempting to solve the trust-region subproblem. In fact, as we will shortly see, additional algorithmic safeguards are important when simultaneously handling trust regions of vastly different sizes.

The main ingredient of our algorithm is that we associate, at iteration  $k$ , a model  $m_{i,k}$  with each element function  $f_i$ . This model is defined on  $\mathcal{R}_i$  in a neighbourhood of the projection of the  $k$ th iterate  $x_k$  onto this subspace and is meant to approximate  $f_i$  for all  $x$  in the elemental trust region

$$\mathcal{B}_j^{(i)} \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid \|P_{\mathcal{R}_i}(x - x_k)\|_k \leq \Delta_{i,k}\}, \quad (10.2.3)$$

where  $\Delta_{i,k} > 0$  is the  $i$ th trust-region radius at iteration  $k$ . In what follows, we will slightly abuse notation by writing  $m_{i,k}(x)$  for an  $x \in \mathbb{R}^n$ , instead of the more complete  $m_{i,k}(P_{\mathcal{R}_i}(x))$ . We use the notation

$$g_{i,k} \stackrel{\text{def}}{=} \nabla_x m_{i,k}(x_k) \in \mathcal{R}_i$$

and refer to the quantity  $\Delta_k^{\min}$  defined by

$$\Delta_k^{\min} \stackrel{\text{def}}{=} \min_{i \in \{1, \dots, p\}} \Delta_{i,k} \quad (10.2.4)$$

as the minimum trust-region radius. Note that, if we were to choose quadratic models for the element functions, these now take the form

$$m_{i,k}(x_k + s) = f_i(x_k) + \langle g_{i,k}, s \rangle + \frac{1}{2} \langle s, H_{i,k} s \rangle,$$

where  $H_{i,k}$  is a symmetric approximation to  $\nabla_{xx} f_i(x_k)$  whose null-space contains the subspace  $\mathcal{N}_i$ . In particular, a Newton model corresponds to the choices  $g_{i,k} = \nabla_x f_i(x_k)$  and  $H_{i,k} = \nabla_{xx} f_i(x_k)$ , which are guaranteed to satisfy this latter condition.

---

entiable functions. Finally, they provide a natural generalization of separable functions as introduced in (10.2.1).

With all the element models at hand, we are now in position to define  $m_k$ , the overall model at iteration  $k$ , whose purpose is to approximate the overall objective function  $f$  in a neighbourhood of the current iterate  $x_k$ . From (10.2.2), it is natural to use the overall partially separable model

$$m_k(x) \stackrel{\text{def}}{=} \sum_{i=1}^p m_{i,k}(x)$$

for all  $x$  in the overall trust region defined by

$$\mathcal{B}_k = \bigcap_{i \in \{1, \dots, p\}} \mathcal{B}_k^{(i)}.$$

Indeed  $\mathcal{B}_k$  is the intersection of all element trust regions, that is, the region in which all element models may be trusted. The actual shape of the trust region  $\mathcal{B}_k$  is determined by the choice of the norm: it corresponds to the intersection of cylinders whose axes are aligned with the subspaces  $\mathcal{N}_i$  and whose radii reflect the quality of the element models: large in subspaces where the element models predict the element functions correctly and smaller in subspaces where the prediction is poorer. The shape of such a trust region (using the Euclidean norm) is shown in Figure 10.2.1. In this figure, we have assumed that the objective function is of the form

$$f(x_1, x_2, x_3) = f_1(x_1, x_2) + f_2(x_1, x_3),$$

which gives that the two range subspaces are

$$\mathcal{R}_1 = \text{span}[e_1, e_2] \quad \text{and} \quad \mathcal{R}_2 = \text{span}[e_1, e_3].$$

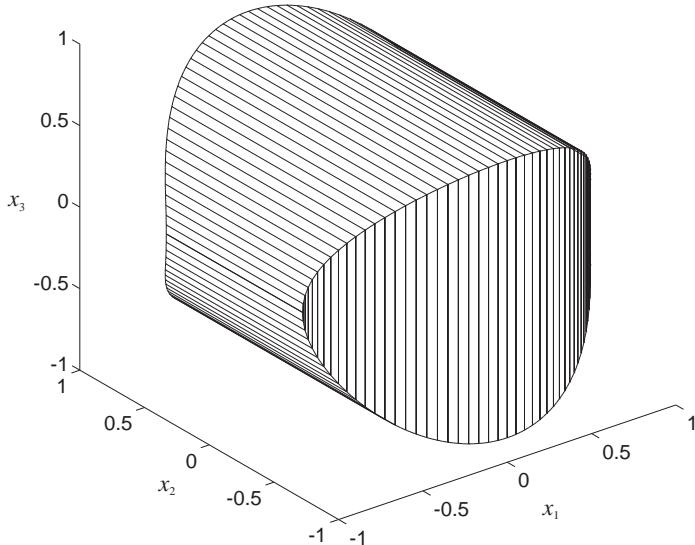


Figure 10.2.1: The shape of a structured trust region using the Euclidean norm.

The current iterate  $x_k$  is at the origin and the two element trust-region radii are chosen to be  $\Delta_{1,k} = 1$  and  $\Delta_{2,k} = 0.85$ . This latter choice is made for ease of illustration, but typically we might expect significantly different values.

In practice, we might wish to choose other norms (see Section 6.7), such as the  $\ell_\infty$  norm. In this latter case, and assuming that the subspaces  $\mathcal{R}_i$  are spanned by subsets of the canonical basis vectors, the shape of the trust region is that of a box, the length of whose sides again reflects the quality of the element models.

We now determine a sufficient decrease of the overall model  $m_k$  within the region  $\mathcal{B}_k$ , whose shape is chosen to reflect the structure of the problem. Special care is needed because this region might be extremely “asymmetric” in the sense that it may allow very large steps in some directions but only very short ones in others. As a consequence, we have to adapt the notion of trust-region “radius” to our new framework and also appropriately reformulate AA.1.

From a practical point of view, we might use a two-stage approach. We first aim to find a step producing a sufficient model decrease in a smaller, but more symmetric, region. Following this, we then allow the step to increase within the trust region while maintaining control over the model decrease. To be specific, let

$$\mathcal{B}_k^{\min} \stackrel{\text{def}}{=} \mathcal{B}_k \cap \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k^{\min}\}$$

be the trust region whose radius<sup>162</sup> is determined by the possibly most nonlinear part of the model. By definition, it is included in  $\mathcal{B}_k$ . Figure 10.2.2, on the next page, illustrates the relation between  $\mathcal{B}_k^{\min}$  and  $\mathcal{B}_k$  in the case of the trust region defined in the Euclidean norm and shown in Figure 10.2.1. In particular, the inclusion  $\mathcal{B}_k^{\min} \subseteq \mathcal{B}_k$  is very apparent.

Applying the conclusions of Section 6.3.4, we may deduce that it is possible to find a step  $s_k^{\min}$  such that  $x_k + s_k^{\min} \in \mathcal{B}_k^{\min}$  and

$$m_k(x_k) - m_k(x_k + s_k^{\min}) \geq \kappa_{\text{mdc}} \|g_k\| \min \left\{ \frac{\|g_k\|}{\beta_k}, \Delta_k^{\min} \right\}$$

for some constant  $\kappa_{\text{mdc}} > 0$ .

However, the restriction that the length of  $s_k^{\min}$  be bounded by  $\Delta_k^{\min}$  makes the whole exercise of shaping  $\mathcal{B}_k$  to reflect the structure of the problem entirely pointless. We might therefore be prepared to accept a larger step provided it remains within the trust region  $\mathcal{B}_k$  and produces a further significant model decrease. More specifically, we allow our algorithm to choose any step  $s_k$  such that  $x_k + s_k \in \mathcal{B}_k$ , and we reformulate AA.1 as follows.

---

<sup>162</sup>Observe that we have retained a single iteration-dependent norm  $\|\cdot\|_k$  for all element trust regions. This is merely for notational convenience, as it would indeed be possible to allow different norms  $\|\cdot\|_{i,k}$  for each element trust region, provided they all remain uniformly equivalent. We do not investigate this generalization here because the  $\Delta_{i,k}$  already provides a means of adapting the trust-region radii to the structure of the problem. Another obvious possibility would be to allow the  $\|\cdot\|_{i,k}$  to vary from element to element while considering a single  $\Delta_k$ .

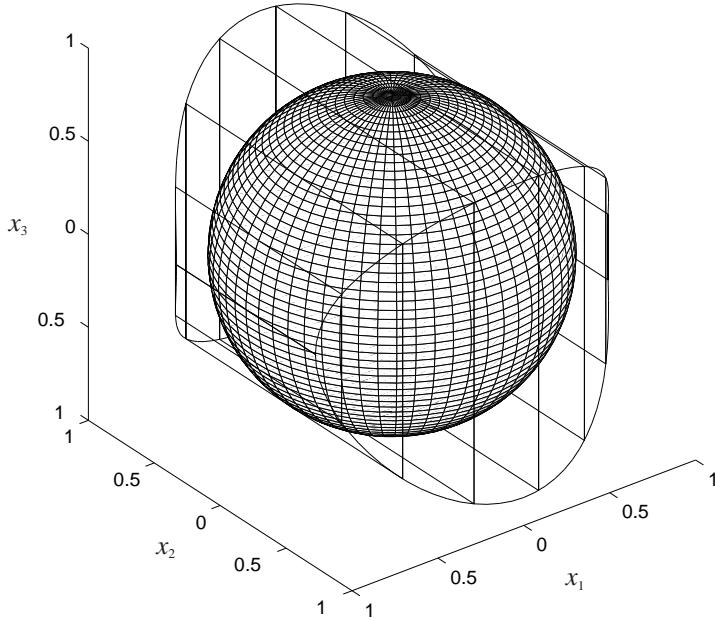


Figure 10.2.2: The relation between  $\mathcal{B}_k^{\min}$  and  $\mathcal{B}_k$ .

**AA.1s** There exists a positive constant<sup>163</sup>  $\kappa_{\text{smd}}$  such that, for all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{smd}} \|g_k\| \min \left\{ \frac{\|g_k\|}{\beta_k}, \max \left[ \Delta_k^{\min}, \frac{\|s_k\|}{\nu_k^S} \right] \right\}.$$

Note that AA.1s holds for  $s_k = s_k^{\min}$  since  $\|s_k^{\min}\| \leq \nu_k^S \Delta_k$ , and hence this condition can be achieved in practice. Observe also that AA.1s is fundamentally different from an angle test of the form

$$|\langle g_k, s_k \rangle| \geq \zeta \|g_k\| \|s_k\| \text{ for some } \zeta \in (0, 1),$$

as it does not prevent  $s_k$  from being orthogonal to the steepest-descent direction, so long as a sufficient model reduction is obtained. This is useful because such a step may occur when moving away from a saddle point of the objective function. Finally note that, as expected, AA.1s reduces to AA.1 in the case where  $f$  has only one element function, since there is only one trust region in this case and thus

$$\Delta_k^{\min} = \Delta_k \geq \frac{\|s_k\|}{\nu_k^S}$$

by construction.

### 10.2.2 A Trust-Region Algorithm Using Problem Structure

We now describe our modification of Algorithm BTR to use the structure of the problem. Besides  $\kappa_{\text{smd}}$  used in AA.1s and the constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  satisfying (6.1.3)

---

<sup>163</sup>“smd” stands for “structured model decrease”.

(p. 116), it depends on the constants

$$\eta_3 \in (\eta_2, 1) \text{ and } 0 < \mu_1 < \mu_2 < 1,$$

whose purpose will soon become clear. In addition, we require a compatibility condition between the  $\eta_i$ 's and the  $\mu_i$ 's. Specifically, we need that

$$\eta_2 - \eta_1 \geq \mu_1 + \mu_2. \quad (10.2.5)$$

Typical values are  $\eta_1 = 0.01$ ,  $\eta_2 = 0.25$ ,  $\eta_3 = 0.75$ ,  $\mu_1 = 0.05$ , and  $\mu_2 = 0.1$ . It is interesting to compare them with (6.1.6) (p. 117).

Aside from the fact that we now have  $p$  models, one for each element, the main difference between our modified algorithm and Algorithm BTR is in the mechanism for updating the trust-region radius (radii). Since the new mechanism is somewhat involved we consider it before stating the complete minimization algorithm.

Three complications arise because of multiple elements. The first is that, although AA.1s ensures that  $m_k(x_k) - m_k(x_k + s_k)$  is always positive, we may not assume in general that the same is true for

$$\delta m_{i,k} \stackrel{\text{def}}{=} m_{i,k}(x_k) - m_{i,k}(x_k + s_k).$$

Because of this difficulty, the intuitive tests on the model fit

$$\frac{\delta f_{i,k}}{\delta m_{i,k}} \geq \eta_j \quad (j = 2, 3), \quad (10.2.6)$$

where

$$\delta f_{i,k} \stackrel{\text{def}}{=} f_i(x_k) - f_i(x_k + s_k)$$

cannot be used directly, because the sign of the denominator is unknown. However, this difficulty can be removed by rewriting the test as

$$\delta f_{i,k} \geq \delta m_{i,k} - (1 - \eta_j)|\delta m_{i,k}| \quad (j = 2, 3), \quad (10.2.7)$$

which is equivalent to (10.2.6) when  $\delta m_{i,k} \geq 0$ . The second complication is that possible cancellation between elements makes it necessary to consider the “accuracy of model fit” for an element relative to the *overall* model fit,

$$\delta m_k \stackrel{\text{def}}{=} m_k(x_k) - m_k(x_k + s_k).$$

Indeed, requiring small relative errors for models with very large values may result in large absolute errors. If  $\delta m_k$  is small (due to cancellation between the  $\delta m_{i,k}$ ), these large errors will then cause  $\delta m_k$  to be a poor prediction of

$$\delta f_k \stackrel{\text{def}}{=} f(x_k) - f(x_k + s_k),$$

and the iteration might be unsuccessful. Therefore, the test (10.2.7) is not suitable either, but should be replaced by

$$\delta f_{i,k} \geq \delta m_{i,k} - (1 - \eta_j)\frac{|\delta m_k|}{p} \quad (j = 2, 3),$$

which measures the fit of the  $i$ th element with respect to the average model change. Observe that this latter condition reduces to the familiar

$$\delta f_k \geq \eta_j \delta m_k \quad (j = 2, 3)$$

when  $p = 1$ . The third complication is that, without further precaution, we could observe a situation where a particular element fits very well and its radius is thus increased, while another element fits badly and its radius is decreased. However, the roles may be reversed at the next iteration and a cycle could occur if no overall progress is made, that is, if the iteration is unsuccessful. To prevent this undesirable cycling, we require that an element trust-region radius not be allowed to increase unless the iteration is successful.

Finally, we remember that our main intent is to allow steps as large as possible. Hence we attempt to reduce an element trust-region radius only when clearly necessary. In this spirit, we first note that, if the model change for an element is negligible, that is, small compared to the overall predicted change  $\delta m_k$ , we do not need to restrict its element trust-region size unless the true element change  $\delta f_{i,k}$  is relatively large compared with the overall predicted change. We therefore try to increase the radii corresponding to negligible elements until they stop being relatively negligible (something which will inevitably arise when convergence occurs). Formally, the distinction between “negligible” elements and “meaningful” ones uses the achieved changes in the element functions and their models defined, for  $i \in \{1, \dots, p\}$ , by  $\delta f_{i,k}$  and  $\delta m_{i,k}$ , respectively. We may then define the set of *negligible* elements at iteration  $k$  as

$$N_k \stackrel{\text{def}}{=} \left\{ i \in \{1, \dots, p\} \mid |\delta m_{i,k}| \leq \frac{\mu_1}{p} \delta m_k \right\} \quad (10.2.8)$$

and the set of *meaningful* elements as its complement  $M_k = \{1, \dots, p\} \setminus N_k$ .

Combining all these considerations, we may now state the mechanism for updating the element trust-region radii. This is Algorithm 10.2.1.

**Algorithm 10.2.1: Structured radii update**

**Step 1: Update the radii of the meaningful elements.** For each  $i \in M_k$ , perform the following.

- If conditions

$$\delta f_{i,k} \geq \delta m_{i,k} - \frac{1 - \eta_3}{p} \delta m_k \quad (10.2.9)$$

and

$$\delta f_k \geq \eta_1 \delta m_k \quad (10.2.10)$$

both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \infty).$$

- If (10.2.9) holds but (10.2.10) fails, then choose

$$\Delta_{i,k+1} = \Delta_{i,k}. \quad (10.2.11)$$

- If (10.2.9) fails but condition

$$\delta f_{i,k} \geq \delta m_{i,k} - \frac{1 - \eta_2}{p} \delta m_k \quad (10.2.12)$$

holds, then choose

$$\Delta_{i,k+1} \in [\gamma_2 \Delta_{i,k}, \Delta_{i,k}].$$

- If condition (10.2.12) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}].$$

**Step 2: Update the radii of the negligible elements.** For each  $i \in N_k$ , perform the following.

- If conditions

$$|\delta f_{i,k}| \leq \frac{\mu_2}{p} \delta m_k, \quad (10.2.13)$$

and (10.2.10) both hold, then choose

$$\Delta_{i,k+1} \in [\Delta_{i,k}, \infty).$$

- If condition (10.2.13) holds but (10.2.10) fails, then choose

$$\Delta_{i,k+1} = \Delta_{i,k}.$$

- If condition (10.2.13) fails, then choose

$$\Delta_{i,k+1} \in [\gamma_1 \Delta_{i,k}, \gamma_2 \Delta_{i,k}].$$

Note that condition (10.2.5) can be viewed as a guarantee that a new iterate will be accepted in (10.2.10) whenever the model reduction obtained for all meaningful elements is also acceptable (i.e., (10.2.12) holds for all  $i \in M_k$ ), irrespective of the contribution of the negligible ones. This interpretation is clarified in Lemma 10.2.2.

Observe again the consistency between the trust-region radii updates in Step 4 of Algorithm BTR and the case where  $p = 1$ . In this latter case, the set  $N_k$  is always empty and (10.2.9) then implies (10.2.10), because of (6.1.3) (p. 116). The update (10.2.11) is thus never applied.

We now state the complete minimization algorithm.

**Algorithm 10.2.2: Structured trust-region algorithm**

**Step 0: Initialization.** The starting point  $x_0$  is given, together with the element function values  $\{f_i(x_0)\}_{i=1}^p$  and the initial trust-region radii  $\{\Delta_{i,0}\}_{i=1}^p$ . The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Set  $k = 0$ .

**Step 1: Model definition.** First choose  $\|\cdot\|_k$ . Then, for  $i \in \{1, \dots, p\}$ , define a model  $m_{i,k}$  in  $\mathcal{B}_k^{(i)}$ .

**Step 2: Step calculation.** Compute a step  $s_k$  such that  $x_k + s_k \in \mathcal{B}_k$  and that satisfies AA.1s.

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{\delta f_k}{\delta m_k}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Element trust-region radii update.** Define the set  $N_k$  of negligible elements according to (10.2.8) and its complement  $M_k$ , the set of meaningful elements. Then apply Algorithm 10.2.1, increment  $k$  by 1 and return to Step 1.

Before starting our global convergence analysis, for future reference we first state some properties that result from the mechanism of Algorithm 10.2.2.

**Lemma 10.2.1** Suppose that  $f$  is partially separable. At each iteration  $k$  of the algorithm,

(i)  $M_k$  contains at least one element. Furthermore

$$\left(1 - \frac{p-1}{p}\mu_1\right)\delta m_k \leq \sum_{i \in M_k} \delta m_{i,k} \leq \left(1 + \frac{p-1}{p}\mu_1\right)\delta m_k; \quad (10.2.14)$$

and

(ii) for all  $i \in \{1, \dots, p\}$ ,

$$\gamma_1 \Delta_{i,k} \leq \Delta_{i,k+1}. \quad (10.2.15)$$

**Proof.** The first result immediately follows from the definition of  $N_k$  and the inequality  $\mu_1 < 1$ . We then deduce that  $N_k$  contains at most  $p-1$  elements. Hence

$$\delta m_k = \sum_{i \in M_k} \delta m_{i,k} + \sum_{i \in N_k} \delta m_{i,k} \leq \sum_{i \in M_k} \delta m_{i,k} + \mu_1 \frac{|N_k|}{p} \delta m_k,$$

from which the first part of (10.2.14) may be deduced. The second inequality in this result is obtained from

$$\sum_{i \in M_k} \delta m_{i,k} = \delta m_k - \sum_{i \in N_k} \delta m_{i,k} \leq \delta m_k + \sum_{i \in N_k} |\delta m_{i,k}|,$$

the relation (10.2.8) and  $|N_k| \leq p - 1$ . The bound (10.2.15) results from the mechanism of Algorithm 10.2.1.  $\square$

We also investigate the coherency between the measure of fit for individual elements and that for the overall model.

**Lemma 10.2.2** Suppose that  $f$  is partially separable and that, at iteration  $k$  of the algorithm, (10.2.12) holds for all  $i \in M_k$  and (10.2.13) holds for all  $i \in N_k$ . Then iteration  $k$  is successful; that is,  $k \in \mathcal{S}$ .

**Proof.** Because (10.2.12) holds for  $i \in M_k$ , we have that

$$\sum_{i \in M_k} \delta f_{i,k} \geq \sum_{i \in M_k} \delta m_{i,k} - (1 - \eta_2) \frac{|M_k|}{p} \delta m_k \geq \left( \eta_2 - \frac{p-1}{p} \mu_1 \right) \delta m_k \quad (10.2.16)$$

for all such  $i$ , where we used the inequality  $|M_k| \leq p$  and Lemma 10.2.1 to deduce the second inequality. On the other hand, since (10.2.13) holds for  $i \in N_k$ , we obtain for these  $i$  that

$$\sum_{i \in N_k} |\delta f_{i,k}| \leq \frac{p-1}{p} \mu_2 \delta m_k, \quad (10.2.17)$$

where we used Lemma 10.2.1 (i) to bound  $|N_k|$ . Now,

$$\delta f_k = \sum_{i \in M_k} \delta f_{i,k} + \sum_{i \in N_k} \delta f_{i,k} \geq \sum_{i \in M_k} \delta f_{i,k} - \sum_{i \in N_k} |\delta f_{i,k}|.$$

Combining this last inequality with (10.2.16) and (10.2.17) gives that

$$\delta f_k \geq \left( \eta_2 - \frac{p-1}{p} \mu_1 - \frac{p-1}{p} \mu_2 \right) \delta m_k,$$

which then yields (10.2.10) because of (10.2.5).  $\square$

Of course, (10.2.12) holds whenever (10.2.9) holds because  $\eta_3 > \eta_2$ . Lemma 10.2.2 therefore shows that (10.2.10) is coherent with the measure of the fit between the element models and element functions.<sup>164</sup>

---

<sup>164</sup>We observe from this proof that the weaker condition

$$\eta_2 - \eta_1 \geq \frac{p-1}{p} (\mu_1 + \mu_2)$$

could be imposed instead of (10.2.5). However, the drawback is that this makes the values of the constants used by the algorithm problem dependent.

### 10.2.3 Convergence Theory

We now turn to the convergence analysis of Algorithm 10.2.2 and show, once more, that the pattern of proof used in Sections 6.4–6.6 can be adapted to this more complicated case.

We first rephrase our assumptions to reflect the partially separable nature of the objective function and model.

**AF.1s** The objective function  $f$  is partially separable and each of its  $p$  element functions  $f_i$  is twice-continuously differentiable on  $\mathbb{R}^n$ .

**AF.3s** The Hessian of each of the  $p$  element functions of  $f$  is uniformly bounded on their respective range subspaces; that is, there exists a positive constant<sup>165</sup>  $\kappa_{\text{ufh}} \geq 1$  such that

$$\|\nabla_{xx} f_i(x)\| \leq \kappa_{\text{ufh}}$$

for all  $x \in \mathcal{R}_i$  and all  $i \in \{1, \dots, p\}$ .

**AF.5s** The gradients of the model's elements are uniformly bounded on their respective range subspaces; that is, there exists a positive constant<sup>166</sup>  $\kappa_{\text{ufg}}$  such that

$$\|\nabla_x f_i(x)\| \leq \kappa_{\text{ufg}}$$

for all  $x \in \mathcal{R}_i$  and all  $i \in \{1, \dots, p\}$ .

**AM.1s** For all  $k$ , the model  $m_k$  is partially separable and each of its  $p$  element functions  $m_{i,k}$  is twice-continuously differentiable on  $\mathcal{B}_k$ .

**AM.2s** The values of the objective function's elements and of the model's elements coincide at the current iterate; that is, for all  $k$  and all  $i \in \{1, \dots, p\}$ ,

$$m_{i,k}(x_k) = f_i(x_k).$$

**AM.3s** The gradient of each of the model's elements at  $x_k$  is equal to the gradient of the corresponding objective function's element; that is, for all  $k$  and all  $i \in \{1, \dots, p\}$ ,

$$g_{i,k} \stackrel{\text{def}}{=} \nabla_x m_{i,k}(x_k) = \nabla_x f_i(x_k).$$

**AM.4s** The Hessian of each of the  $p$  model's elements remains bounded within the trust region; that is, there exists a constant  $\kappa_{\text{umh}} \geq 1$  such that

$$\|\nabla_{xx} m_{i,k}(x)\| \leq \kappa_{\text{umh}} - 1 \quad \text{for all } x \in \mathcal{B}_k$$

for all  $k$  and all  $i \in \{1, \dots, p\}$ .

<sup>165</sup>“ufh” stands for “upper bound on the function's element Hessians”.

<sup>166</sup>“ufg” stands for “upper bound on the function's element gradients”.

Note that AF.2 does not need rephrasing, as it is (fortunately) not necessary that each of the element functions be bounded below, but merely that their sum  $f$  is. We shall discuss AF.5s, the requirement that the element gradients must be uniformly bounded, below.

We now examine the results of Sections 6.4–6.6 in sequence and discuss their adaptation in detail, when AF.1, AF.3, AM.1–AM.4, and AA.1 are replaced by AF.1s, AF.3s, AM.1s–AM.4s, and AA.1s, respectively. Theorem 6.4.1 (p. 133) is now valid for each element separately.

**Theorem 10.2.3** Suppose that AF.1s, AF.3s, AN.1, and AM.1s–AM.4s hold. Then, we have that

$$|f_i(x_k + s_k) - m_{i,k}(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_{i,k}^2 \quad (10.2.18)$$

for all  $i \in \{1, \dots, p\}$  and all  $k$ , where  $\kappa_{\text{ubh}} \stackrel{\text{def}}{=} \kappa_{\text{une}}^2 \max[\kappa_{\text{ufh}}, \kappa_{\text{umh}}] \geq 1$ .

We then prove a simple technical result relating the size of the model change for the  $i$ th element to that of the projection of the step onto the corresponding range subspace.

**Lemma 10.2.4** Suppose that AF.5s, AN.1, and AM.1s–AM.4s hold. Consider iteration  $k$  of Algorithm 10.2.2 and suppose that, for some  $i \in \{1, \dots, p\}$ ,

$$\kappa_{\text{une}} \Delta_{i,k} \leq 1. \quad (10.2.19)$$

Then we have that

$$|\delta m_{i,k}| \leq \kappa_{\text{uem}} \Delta_{i,k} \quad (10.2.20)$$

for some constant<sup>167</sup>  $\kappa_{\text{uem}} > 0$  independent of  $i$  and  $k$ .

**Proof.** We first note that (10.2.4) and (10.2.19) imply that

$$\kappa_{\text{une}} \Delta_k^{\min} \leq 1. \quad (10.2.21)$$

Using AM.1s–AM.4s and the mean value theorem, we also obtain, for some  $\zeta \in [x_k, x_k + s_k]$ , that

$$\begin{aligned} |\delta m_{i,k}| &\leq |\langle \nabla_x f_i(x_k), s_k \rangle| + \frac{1}{2} |\langle s_k, \nabla_{xx} m_{i,k}(\zeta) s_k \rangle| \\ &= |\langle \nabla_x f_i(x_k), P_{\mathcal{R}_i}(s_k) \rangle| + \frac{1}{2} |\langle P_{\mathcal{R}_i}(s_k), \nabla_{xx} m_{i,k}(\zeta) P_{\mathcal{R}_i}(s_k) \rangle|. \end{aligned}$$

Now combining (10.2.3), (10.2.19), and (10.2.21), we can deduce, using the Cauchy–Schwarz inequality, AM.4s, AF.5s, and AN.1, that

$$|\delta m_{i,k}| \leq \kappa_{\text{ufg}} \|P_{\mathcal{R}_i}(s_k)\| + \frac{1}{2} \kappa_{\text{umh}} \|P_{\mathcal{R}_i}(s_k)\|^2 \leq \kappa_{\text{une}} [\kappa_{\text{ufg}} + \frac{1}{2} \kappa_{\text{umh}}] \Delta_{i,k}. \quad (10.2.22)$$

<sup>167</sup>“uem” stands for “upper bound on element model”.

Inequality (10.2.22) then gives (10.2.20) with  $\kappa_{\text{uem}} \stackrel{\text{def}}{=} \kappa_{\text{une}}(\kappa_{\text{ufg}} + \frac{1}{2}\kappa_{\text{umh}})$ .  $\square$

We next prove the equivalent of Theorem 6.4.3 (p. 135) directly, absorbing the analog of Theorem 6.4.2 (p. 134) into the proof.

**Theorem 10.2.5** Suppose that AF.1s–AF.3s, AN.1, AM.1s–AM.4s, and AA.1s hold. Suppose furthermore that there exists a constant  $\kappa_{\text{lbg}} > 0$  such that

$$\|g_k\| \geq \kappa_{\text{lbg}} \quad (10.2.23)$$

for all  $k$ . Then there is a constant  $\kappa_{\text{lbd}} > 0$  such that, for all  $k$ ,

$$\Delta_k^{\min} \geq \kappa_{\text{lbd}}.$$

**Proof.** In order to derive a contradiction, we assume that there exists a  $k$  such that

$$\Delta_k^{\min} < \gamma_1 \min \left[ \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubh}}}, \kappa_{\text{une}}, \frac{\mu_1 \epsilon^2 (1 - \eta_3)}{\kappa_{\text{ubh}} \kappa_{\text{uem}} p^2}, \frac{\epsilon(\mu_2 - \mu_1)}{\kappa_{\text{ubh}} p}, \Delta_0^{\min} \right] \stackrel{\text{def}}{=} \kappa_{\text{lbd}}, \quad (10.2.24)$$

where  $\epsilon \stackrel{\text{def}}{=} \gamma_1 \kappa_{\text{smd}} \kappa_{\text{lbg}}$ . Now define  $r$  to be the smallest iteration number such that (10.2.24) holds. We remark that  $r \geq 1$  because  $\Delta_r^{\min} < \gamma_1 \Delta_0^{\min}$  and the inequality  $\gamma_1 < 1$ . Also fix  $i$  such that  $\Delta_r^{\min} = \Delta_{i,r}$ . The bounds (10.2.15) and (10.2.24) then ensure that

$$\Delta_{r-1}^{\min} \leq \Delta_{i,r-1} \leq \frac{\Delta_{i,r}}{\gamma_1} \leq \frac{\kappa_{\text{lbd}}}{\gamma_1} \leq \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubh}}}. \quad (10.2.25)$$

We note that the definitions of  $i$  and  $r$  give that

$$\Delta_{i,r} = \Delta_r^{\min} < \Delta_{r-1}^{\min}.$$

Using this inequality with AA.1s, (10.2.23), and the inequalities  $\beta_{r-1} \leq \kappa_{\text{ubh}}$  and (10.2.25), we obtain that

$$\begin{aligned} \delta m_{r-1} &\geq \kappa_{\text{smd}} \kappa_{\text{lbg}} \min \left\{ \frac{\kappa_{\text{lbg}}}{\beta_{r-1}}, \max[\Delta_{r-1}^{\min}, \|s_{r-1}\|] \right\} \\ &\geq \kappa_{\text{smd}} \kappa_{\text{lbg}} \min \left\{ \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubh}}}, \Delta_{r-1}^{\min} \right\} \\ &\geq \kappa_{\text{smd}} \kappa_{\text{lbg}} \Delta_{r-1}^{\min} \\ &\geq \kappa_{\text{smd}} \kappa_{\text{lbg}} \Delta_{i,r}, \end{aligned}$$

which ensures, because of (10.2.15) and the definition of  $\epsilon$ , that

$$\delta m_{r-1} \geq \epsilon \Delta_{i,r-1}. \quad (10.2.26)$$

But (10.2.24) and (10.2.25) imply that

$$\kappa_{\text{une}} \Delta_{i,r-1} \leq \frac{\kappa_{\text{une}} \kappa_{\text{lbd}}}{\gamma_1} \leq 1.$$

We may thus apply Lemma 10.2.4 and deduce that

$$|\delta m_{i,r-1}| \leq \kappa_{\text{uem}} \Delta_{i,r-1} \leq \frac{\kappa_{\text{uem}}}{\epsilon} \delta m_{r-1}, \quad (10.2.27)$$

where we also used (10.2.26).

Suppose first that  $i \in M_{r-1}$ , which guarantees that  $\delta m_{i,r-1} \neq 0$ . Then, using (10.2.8) and (10.2.26), we have that

$$|\delta m_{i,r-1}| > \frac{\mu_1}{p} \delta m_{r-1} \geq \frac{\mu_1 \epsilon}{p} \Delta_{i,r-1}. \quad (10.2.28)$$

Because of AM.2s, (10.2.18), and (10.2.28), we therefore obtain that

$$\left| \frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1 \right| = \frac{|f_i(x_{r-1} + s_{r-1}) - m_{i,r-1}(x_{r-1} + s_{r-1})|}{|\delta m_{i,r-1}|} \leq \frac{\kappa_{\text{ubh}} p}{\mu_1 \epsilon} \Delta_{i,r-1}. \quad (10.2.29)$$

But (10.2.24) and (10.2.25) together give that

$$\Delta_{i,r-1} \leq (1 - \eta_3) \frac{\mu_1 \epsilon^2}{\kappa_{\text{ubh}} \kappa_{\text{uem}} p^2},$$

which, with (10.2.29), implies that

$$\left| \frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} - 1 \right| \leq \frac{(1 - \eta_3) \epsilon}{\kappa_{\text{uem}} p}. \quad (10.2.30)$$

Consider first the case where  $\delta m_{i,r-1} > 0$ . We may then apply (10.2.27) and deduce that

$$\begin{aligned} \delta m_{i,r-1} - \frac{1 - \eta_3}{p} \delta m_{r-1} &= \delta m_{i,r-1} \left( 1 - \frac{(1 - \eta_3)}{p} \frac{\delta m_{r-1}}{|\delta m_{i,r-1}|} \right) \\ &\leq \delta m_{i,r-1} \left( 1 - \frac{(1 - \eta_3) \epsilon}{\kappa_{\text{uem}} p} \right). \end{aligned} \quad (10.2.31)$$

Using (10.2.30), we now deduce that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \geq 1 - \frac{(1 - \eta_3) \epsilon}{\kappa_{\text{uem}} p},$$

and therefore, because of (10.2.31), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1} \left( 1 - \frac{(1 - \eta_3) \epsilon}{\kappa_{\text{uem}} p} \right) \geq \delta m_{i,r-1} - \frac{1 - \eta_3}{p} \delta m_{r-1},$$

which implies that (10.2.9) holds for element  $i$  at iteration  $r - 1$ .

Now turn to the case where  $\delta m_{i,r-1} < 0$ . Because of (10.2.27), we deduce that

$$\begin{aligned} \delta m_{i,r-1} - \frac{1 - \eta_3}{p} \delta m_{r-1} &= \delta m_{i,r-1} \left( 1 + \frac{(1 - \eta_3)}{p} \frac{\delta m_{r-1}}{|\delta m_{i,r-1}|} \right) \\ &\leq \delta m_{i,r-1} \left( 1 + \frac{(1 - \eta_3) \epsilon}{\kappa_{\text{uem}} p} \right). \end{aligned} \quad (10.2.32)$$

As above, we use (10.2.30) to obtain that

$$\frac{\delta f_{i,r-1}}{\delta m_{i,r-1}} \leq 1 + \frac{(1 - \eta_3)\epsilon}{\kappa_{\text{uem}} p},$$

and therefore, because of (10.2.32), that

$$\delta f_{i,r-1} \geq \delta m_{i,r-1} \left( 1 + \frac{(1 - \eta_3)\epsilon}{\kappa_{\text{uem}} p} \right) \geq \delta m_{i,r-1} - \frac{1 - \eta_3}{p} \delta m_{r-1},$$

which again implies that (10.2.9) holds for element  $i$  at iteration  $r - 1$ .

Suppose now that  $i \in N_{r-1}$ . Then, because of AM.2s, (10.2.8), and (10.2.18), we have that

$$\begin{aligned} |\delta f_{i,r-1}| &\leq |\delta m_{i,r-1}| + |f_i(x_{r-1} + s_{r-1}) - m_{i,r-1}(x_{r-1} + s_{r-1})| \\ &\leq \frac{\mu_1}{p} \delta m_{r-1} + \kappa_{\text{ubh}} \Delta_{i,r-1}^2. \end{aligned} \quad (10.2.33)$$

Now, multiplying (10.2.26) by  $\Delta_{i,r-1}$ , we obtain that

$$\Delta_{i,r-1}^2 \leq \frac{\Delta_{i,r-1}}{\epsilon} \delta m_{r-1}. \quad (10.2.34)$$

Combining (10.2.33) and (10.2.34), we deduce that

$$|\delta f_{i,r-1}| \leq \left( \frac{\mu_1}{p} + \frac{\kappa_{\text{ubh}}}{\epsilon} \Delta_{i,r-1} \right) \delta m_{r-1}. \quad (10.2.35)$$

Observing now that (10.2.24) and (10.2.25) imply that

$$\Delta_{i,r-1} \leq \frac{\epsilon(\mu_2 - \mu_1)}{\kappa_{\text{ubh}} p},$$

we obtain from (10.2.35) that

$$|\delta f_{i,r-1}| \leq \frac{\mu_2}{p} \delta m_{r-1}.$$

But this inequality implies that (10.2.13) holds for element  $i$  at iteration  $r - 1$ .

Thus either (10.2.9) or (10.2.13) holds for the  $i$ th element at iteration  $r - 1$ , and the mechanism of Algorithm 10.2.1 then implies that  $\Delta_{i,r} \geq \Delta_{i,r-1}$ . But we may deduce from this inequality and the definition of  $\Delta_{r-1}^{\min}$  in (10.2.4) that

$$\Delta_{r-1}^{\min} \leq \Delta_{i,r-1} \leq \Delta_{i,r},$$

which contradicts the assumption that  $r$  is the smallest iteration number for which (10.2.24) holds. The inequality (10.2.24) therefore never holds and we obtain the desired bound.  $\square$

The role of AF.5s is clear in the above proof. Because it allows for the bound (10.2.20), AF.5s ensures the inequality

$$\frac{\delta m_{r-1}}{|\delta m_{i,r-1}|} \leq \frac{\epsilon}{\kappa_{\text{uem}}}.$$

This says that the changes in the overall model and that of the  $i$ th element are comparable. Note that the proof only requires  $\|\nabla_x f_i(x_k)\|$  to be bounded, a property that is automatically ensured if the sequence of iterates  $\{x_k\}$  produced by Algorithm 10.2.2 remains in a bounded domain, which is a very common occurrence.

Theorem 6.4.4 (p. 136) may then be extended in a straightforward manner.

**Theorem 10.2.6** Suppose that AF.1s, AF.3s, AF.5s, AM.1s–AM.4s, and AA.1s hold. Suppose furthermore that there are only finitely many successful iterations. Then  $x_k = x_*$  for all sufficiently large  $k$ , and  $x_*$  is first-order critical.

**Proof.** The mechanism of the algorithm ensures that  $x_* = x_{k_0+1} = x_{k_0+j}$  for all  $j > 0$ , where  $k_0$  is the index of the last successful iterate. Note now that Lemma 10.2.2 implies that, if  $k \notin \mathcal{S}$ , then (10.2.12) or (10.2.13) must be violated for at least one element. Since no  $\Delta_{i,k}$  can be increased at unsuccessful iterations, we deduce that  $\Delta_k^{\min}$  converges to zero. This contradicts Lemma 10.2.5 unless  $\|g_{k_0+1}\| = 0$  and  $x_*$  is first-order critical.  $\square$

Theorem 6.4.5 (p. 136) can be adapted to our context even more easily. In its proof, it is enough to replace  $\Delta_k$  by  $\Delta_k^{\min}$  and to observe that AA.1s implies AA.1. This yields that

$$\delta f_k \geq \eta_1 \delta m_k \geq \eta_1 \kappa_{\text{smd}} \epsilon \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}} \right],$$

which plays the role of (6.4.17). The rest of the proof is unmodified. The proof of Theorem 6.4.6 (p. 137) is extended to cover Algorithm 10.2.2 in the same way. We therefore conclude that all limit points of the sequence of iterates generated by Algorithm 10.2.2 are first-order critical.

The analogs of the results of Section 6.5 (p. 146) follow without any modification. Similarly, it is easy to verify that Theorems 6.6.3–6.6.5 and 6.6.7 (pp. 154–157) remain valid provided the adaptations mentioned in the previous paragraph for the proof of Theorem 6.4.5 are made, and if AA.2 is rephrased as follows.

**AA.2s** If  $\tau_k = \lambda_{\min}[\nabla_{xx} m_k(x_k)] < 0$ , then, for some constant  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$ ,

$$\delta m_k \geq \kappa_{\text{sod}} |\tau_k| \min \left[ \tau_k^2, [\nu_k^E]^2 [\Delta_k^{\min}]^2 \right].$$

Whether a result equivalent to Theorem 6.6.8 (p. 159) is true for Algorithm 10.2.2 remains, for now, an open question. The main difficulty is to provide a generalization of AA.3 that is compatible with the element by element technique used in Algorithm 10.2.1.

Numerical experiments have shown that the performance of Algorithm 10.2.2 is usually very similar to that of Algorithm BTR, although the former significantly outperforms the latter in a few instances. However, the experience with Algorithm 10.2.2

is to date very limited, and further investigation is needed to see if this modification of a trust-region algorithm will prove efficient in practice and justify the slight additional complication of the method. The real power of the concept may ultimately appear when minimizing augmented Lagrangians or other penalty-like functions, because scaling is much more critical there than in many of the classical unconstrained test examples.

## Notes and References for Section 10.2

The concept of structured trust regions in relation to the partially separable decomposition of an objective function was first considered by Conn, Gould, Sartenaer, and Toint (1996b), albeit in the more general context of minimizing a partially separable function subject to simple bound constraints and inexact derivatives (techniques similar to those of Section 8.4 are used). In our exposition, we have restricted ourselves to the unconstrained case and have supposed that exact derivatives are available. The idea of structured trust regions was further refined in the Ph.D. thesis of Shahabuddin (1996) in a slightly simpler setting: the concept of negligible elements is not used, and the algorithms proposed are restricted to the unconstrained case. Finally, we mention Nelson and Papalambros (1998, 1999) and Alexandrov and Dennis (1999) for other approaches using trust-region methods for structured problems.

## 10.3 Trust Regions and Linesearches

### 10.3.1 Linesearch Algorithms as Trust-Region Methods

We next consider a somewhat extreme but interesting application of our convergence analysis, namely, the special case in which we choose

$$m_k(x_k + s) = f(x_k + s)$$

for all  $s$  such that  $\|s\|_k \leq \Delta_k$ , that is, if we choose the model to be the objective function itself. In this case, it is clear that

$$\rho_k = 1$$

for all  $k$ , where  $\rho_k$  is the ratio of achieved to predicted reduction defined in (6.1.4) (p. 116). All iterations are thus successful, and the trust region  $\Delta_k$  never decreases. If we suppose that our iterates remain in a bounded domain of  $\mathbb{R}^n$  and  $\Delta_0$  is chosen large enough, then  $\Delta_k$  becomes irrelevant in the algorithm, in the sense that the constraint  $\|s_k\|_k \leq \Delta_k$  is never active. We therefore obtain an algorithm that only depends on the Cauchy and eigenpoints, they themselves being determined by linesearches along the steepest-descent and negative curvature directions, provided the latter exist.

It is not difficult to adapt the Armijo linesearch techniques (6.3.17), (6.3.18) (p. 128) and (6.6.8), (6.6.9) (p. 150) to the case where  $\Delta_k$  is essentially infinite. The first step is to redefine a (theoretically arbitrary) initial stepsize corresponding to  $j = 0$ . In the

case of the search for the Cauchy point, (6.3.17) may simply be replaced by

$$x_k(j) = x_k - \kappa_{\text{bck}}^j g_k.$$

We then calculate  $j_c$  as the smallest (possibly negative) integer such that (6.3.18) is satisfied for  $j = j_c$ , but violated for  $j = j_c - 1$ . Note that  $j_c$  must be bounded below if the level set  $\mathcal{L}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$  is bounded. Figure 10.3.1 illustrates a situation where  $\kappa_{\text{bck}} = 0.75$  and  $j_c = -1$ . If negative curvature is present in  $\nabla_{xx} m_k(x_k) = \nabla_{xx} f(x_k)$ , we may determine an approximate eigenvector  $u_k$  satisfying (6.6.7) (p. 150), except that  $u_k$  is now scaled to 1. As for the Cauchy point,  $j_e$  is computed as the smallest (possibly negative) integer such that (6.6.9) holds for  $j = j_e$ , but is violated for  $j = j_e - 1$ . In effect, both linesearch schemes are modified to start with an arbitrary initial stepsize and allow not only backtracking but also extending this initial stepsize, if necessary. Of course, it may be practically more efficient to pursue the minimization of the objective function beyond the Cauchy point (or beyond the eigenpoint), for instance using second-order information, but any subsequent decrease of the objective is theoretically covered by the fact that we have always allowed (and even encouraged)  $m_k(x_k + s_k)$  to be strictly less than  $m_k(x_k^C)$  or  $m_k(x_k^E)$  (see condition (6.6.10) [p. 151]). Maybe more realistically, we may wish to compute a modified Cauchy arc as in Section 8.1.5. For instance, we may wish to compute a direction  $d_k$  that satisfies the conditions

$$\langle g_k, d_k \rangle \leq -\kappa_{\text{sld}} \|g_k\|^2 \text{ and } \|d_k\| \leq \kappa_{\text{nrd}} \|g_k\|, \quad (10.3.1)$$

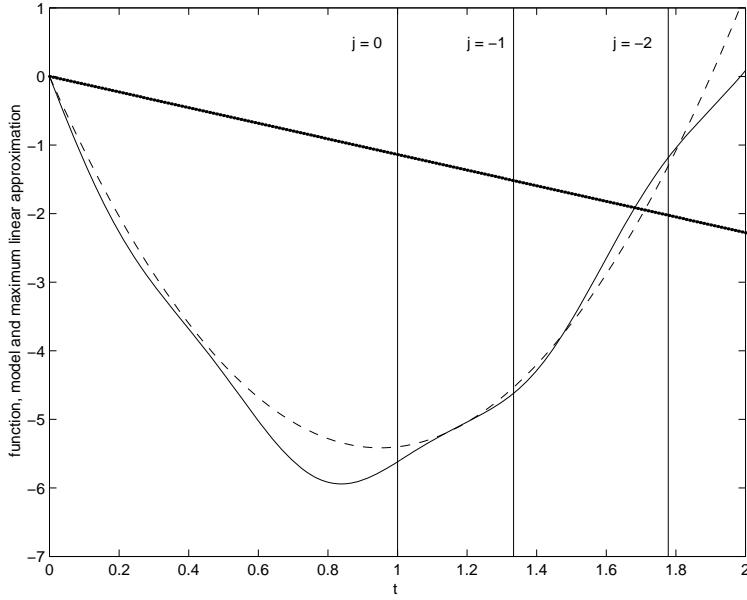


Figure 10.3.1: The objective function (continuous line), its quadratic model (dashed), and the maximum linear approximation (thick) for an example of a linesearch with arbitrary initial step where  $j_c = -1$ .

where  $\kappa_{\text{sld}} \leq \kappa_{\text{nd}}$  are positive constants<sup>168</sup> independent of  $k$ . There are many ways in which such a direction can be calculated. For example,  $d_k$  may be chosen as the model Newton direction  $-[\nabla_{xx}m_k(x_k)]^{-1}g_k$ , provided  $\nabla_{xx}m_k(x_k)$  is positive definite and has a bounded condition number. More generally, we may consider the case where

$$d_k = -B_k^{-1}g_k$$

for some symmetric positive definite matrix  $B_k$  such that

$$\kappa_{\text{lbb}} \leq \lambda_{\min}[B_k] \leq \lambda_{\max}[B_k] \leq \kappa_{\text{ubb}},$$

where  $\kappa_{\text{ubb}} \geq \kappa_{\text{lbb}} > 0$  are constants<sup>169</sup> independent of  $k$ . We then obtain that

$$\|B_k^{-1}g_k\| \leq \|B_k^{-1}\| \|g_k\| = \frac{1}{\lambda_{\min}[B_k]} \|g_k\| \leq \frac{1}{\kappa_{\text{lbb}}} \|g_k\|$$

and

$$\langle g_k, B_k^{-1}g_k \rangle \geq \lambda_{\min}[B_k^{-1}] \|g_k\|^2 = \frac{1}{\lambda_{\max}[B_k]} \|g_k\|^2 \geq \frac{1}{\kappa_{\text{ubb}}} \|g_k\|^2,$$

and conditions (10.3.1) are satisfied for such a  $d_k$ .

The resulting algorithmic framework is then uniquely based on these modified line-searches. We call this algorithm the *double linesearch algorithm*: it is formally summarized in Algorithm 10.3.1.

#### Algorithm 10.3.1: Double linesearch algorithm

**Step 0: Initialization.** An initial  $x_0$  is given. Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Determination of the search directions.** Determine a direction  $d_k$  satisfying (10.3.1). If  $\tau_k$ , the leftmost eigenvalue of  $\nabla_{xx}f(x_k)$ , is negative, also determine a direction  $u_k$  such that

$$\langle u_k, \nabla_x f(x_k) \rangle \leq 0, \quad \|u_k\| = 1, \quad \text{and} \quad \langle u_k, \nabla_{xx}m_k(x_k)u_k \rangle \leq \kappa_{\text{snc}}\tau_k$$

for some  $\kappa_{\text{snc}} \in (0, 1]$ .

**Step 2: Linesearches.** Determine the modified Cauchy point  $x_k^{\text{MC}}$  using the line-search with arbitrary initial stepsize described above, applied to the objective function. If  $u_k$  has been calculated, similarly determine the eigenpoint  $x_k^{\text{E}}$ .

**Step 3: Select a new point.** Compute a new point  $x_{k+1}$  such that

$$f(x_k) - f(x_{k+1}) \geq \kappa_{\text{nmd}} \{f(x_k) - \min[f(x_k^{\text{C}}), f(x_k^{\text{E}})]\} \quad (10.3.2)$$

for some  $\kappa_{\text{nmd}} \in (0, 1]$ . Increment  $k$  by 1 and go to Step 1.

<sup>168</sup>“sld” stands for “slope along  $\underline{d}$ ”, “nd” for “norm of  $\underline{d}$ ”.

<sup>169</sup>“lbb” stands for “lower bound on  $\underline{B}$ ”, “ubb” for “upper bound on  $\underline{B}$ ”.

Figure 10.3.2 shows two possible Step 2s for Algorithm 10.3.1. It is perhaps surprising that its underlying convergence theory, including convergence to second-order critical points, as we will now show, can be derived from the analysis of trust-region methods.

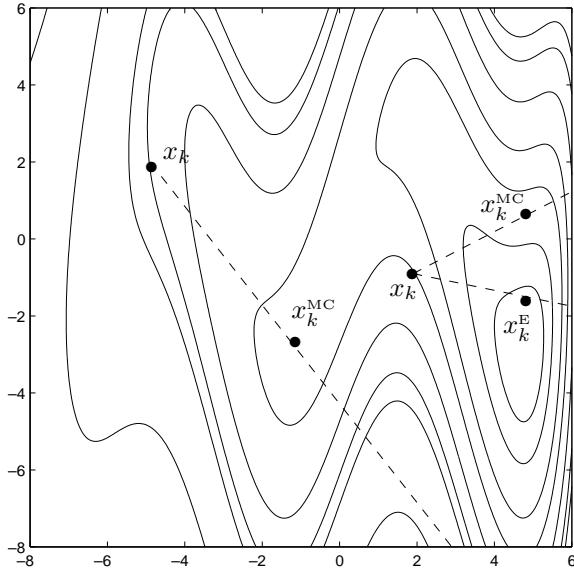


Figure 10.3.2: Two possible iterations of the linesearch-based algorithm: An iteration corresponding to a (local) convex model is shown on the left, while an iteration involving the determination of an eigenpoint is pictured on the right.

**Theorem 10.3.1** Suppose that AF.1–AF.3 hold. Suppose furthermore that the level set

$$\mathcal{L}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$$

is bounded and that there exists a constant  $\kappa_{\text{lch}} > 0$  such that

$$\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\| \leq \kappa_{\text{lch}} \|x - y\| \quad (10.3.3)$$

for all  $x, y \in \mathcal{L}_0$ . Then the conclusions of Theorems 6.4.6 (p. 137), 6.5.2 (p. 141), 6.5.5 (p. 146), 6.6.4 (p. 154), 6.6.5 (p. 155), and 6.6.7 (p. 157) remain valid.

**Proof.** In order to prove this result, it is enough to cast Algorithm 10.3.1 in the framework of Algorithm BTR (p. 116). To do this, we choose an initial trust-region radius  $\Delta_0$  in Algorithm BTR that is large enough to ensure that

$$\Delta_0 \geq \max_{x, y \in \mathcal{L}_0} \|x - y\| \quad \text{and} \quad \Delta_0 \geq -\min_{x \in \mathcal{L}_0} \lambda_{\min}[\nabla_{xx} f(x)],$$

and define

$$m_k(x_k + s) = f(x_k + s) \quad (10.3.4)$$

for all  $k$ . As a consequence,  $\rho_k = 1$  and  $\Delta_k \geq \Delta_0$  for all  $k$ . Furthermore, since  $m_k(x_k^{\text{MC}}) = f(x_k^{\text{MC}}) < f(x_k) \leq f(x_0)$  and, when there is negative curvature,  $m_k(x_k^{\text{E}}) = f(x_k^{\text{E}}) < f(x_k) \leq f(x_0)$ , it follows that  $x_k^{\text{MC}}$ ,  $x_k^{\text{E}}$ , and thus  $x_{k+1}$  always remain in the interior of the trust region for all  $k$ . Note that the choice of the constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  is irrelevant here, and the resulting version of Algorithm BTR is identical to the double linesearch algorithm, that is, Algorithm 10.3.1. Furthermore, the choice of the search direction  $d_k$  satisfying (10.3.1) can be viewed as a special case of the modified Cauchy point discussed in Section 8.1.5. This is the case since (10.3.1) clearly implies that

$$|\langle g_k, d_k \rangle| \geq \frac{\kappa_{\text{sld}}}{\kappa_{\text{nrd}}} \|g_k\| \|d_k\|,$$

which corresponds to (8.1.48) (p. 268), with  $\epsilon(x) = \kappa_{\text{sld}} \kappa_{\text{nrd}}^{-1} \in [0, 1]$  for all  $x$ . It therefore only remains to verify that AM.1–AM.6, AA.1b, and AA.2 also hold. But AM.1–AM.4 and AM.5 result from AF.1–AF.3 and (10.3.4), while AM.6 is implied by (10.3.3), and AA.1b results from (10.3.2) and Theorem 8.1.10 (p. 267). Finally, AA.2 results from (10.3.2), (10.3.3), and Theorem 6.6.2 (p. 151).  $\square$

### Notes and References for Subsection 10.3.1

The fact that linesearch algorithms can be considered as a special case of trust-region methods was independently recognized by Shultz, Schnabel, and Byrd (1985) and Toint (1988), although in different contexts. Our exposition follows the latter more general framework. Algorithm 10.3.1 is reminiscent of Gould, Lucidi, Roma, and Toint (1998), where only one of the two linesearches is performed according to a test measuring their associated expected model improvement. The link between linesearch and trust-region methods is also explored in Vicente (1996).

#### 10.3.2 Backtracking at Unsuccessful Iterations

We now turn to a second application of our modified definitions for the Cauchy point and arc of Section 8.1.5. We motivate this by observing that, if iteration  $k$  is unsuccessful, a new step must be recomputed from  $x_{k+1} = x_k$ , possibly with the same model that we used at iteration  $k$ . In particular, this is the case when the model  $m_k$  is the Newton model, given by the first three terms of the Taylor series of the objective function at  $x_k$ . Since a suitable step was already computed using this information at iteration  $k$  within a trust region of radius  $\Delta_k$ , it is natural to ask whether the computational effort invested in this computation could be reused at iteration  $k + 1$ , where a step must be computed for the same model but within a trust region of radius  $\Delta_{k+1} < \Delta_k$ .

To answer this question, we suppose, for now without further justification, that the unsuccessful step  $s_k$  is gradient related in the sense of (8.1.48) (p. 268). In this

case, we may exploit the knowledge of  $s_k$  to define a new step  $s_{k+1}$  as follows. We recompute a (modified) Cauchy point  $x_{k+1}^C$  from the iterate  $x_{k+1} = x_k$  along the direction  $d_{k+1} = s_k$ , using the backtracking linesearch procedure of Theorem 8.1.10 (p. 267) and the updated radius  $\Delta_{k+1} \leq \gamma_2 \Delta_k$ . We then set  $s_{k+1} = x_{k+1}^C - x_{k+1}$ , as shown in Figure 10.3.3.

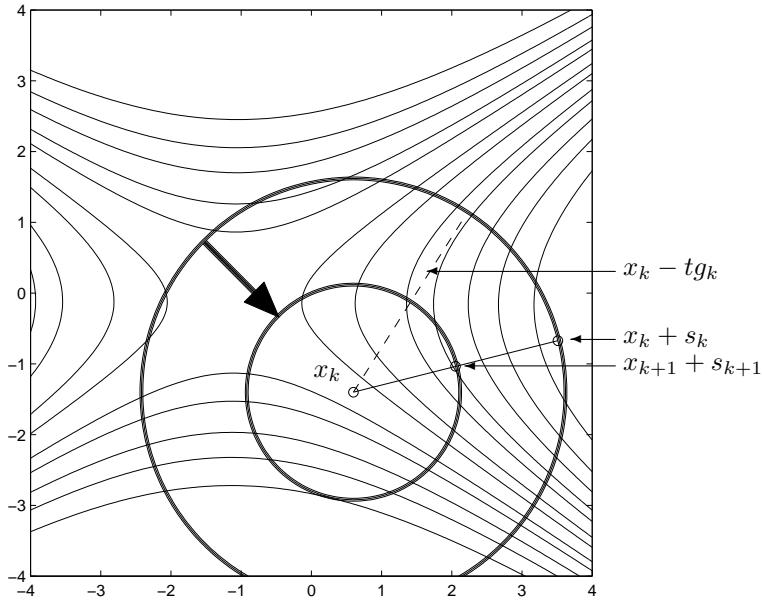


Figure 10.3.3: Backtracking along  $s_k$  when  $x_k + s_k$  is unsuccessful and the trust-region radius shrinks.

This leads to the version of the basic algorithm including these backtracking steps, which we define as Algorithm 10.3.2.

**Algorithm 10.3.2: Backtracking trust-region algorithm**

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116), as are  $\kappa_{\text{bek}} \in (0, 1)$  and  $\kappa_{\text{ubs}} \in (0, \frac{1}{2})$ . An upper bound  $\Delta_{\max} \geq \Delta_0$  is also given. Compute  $f(x_0)$  and set  $k = 0$  and  $\mathcal{S} = \{-1\}$ .

**Step 1: Combined model definition and step calculation.**

Choose  $\|\cdot\|_k$ .

**Step 1a: After a successful iteration.** If  $k - 1 \in \mathcal{S}$ , define a model  $m_k$  in  $\mathcal{B}_k$  and compute a step  $s_k$  that “sufficiently reduces the model”  $m_k$  in the sense of AA.1/AA.2, which is gradient related in the sense of (8.1.48) (p. 268), and such that  $x_k + s_k \in \mathcal{B}_k$ .

**Step 1b: After an unsuccessful iteration.** If  $k - 1 \notin \mathcal{S}$ , set  $m_k = m_{k-1}$ ; find  $j_c$ , the smallest nonnegative integer  $j$  such that

$$f(x_k + \kappa_{\text{bck}}^{j_c} d_k) \leq f(x_k) + \kappa_{\text{ubs}} \kappa_{\text{bck}}^j \langle g_k, d_k \rangle, \quad (10.3.5)$$

where  $d_k = \Delta_k \frac{s_{k-1}}{\|s_{k-1}\|_k}$ ; set  $s_k = \kappa_{\text{bck}}^{j_c} d_k$ ; and reset  $\Delta_k = \|s_k\|_k$ .

**Step 2: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$  and add  $\{k\}$  to  $\mathcal{S}$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 3: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \Delta_{\max}] & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

Besides the backtracking procedure, which we have just described and which comprises Step 1b, we have further modified Algorithm BTR to impose a uniform upper bound  $\Delta_{\max}$  on the trust-region radius. We will see below why this condition is important in theory.

We also note that, in many cases, the value  $j_c = 0$  is quite likely. Indeed, it often happens that (10.3.5) holds at  $x_k + ts_k$  for all  $t \in (0, 1]$ , in which case the only purpose of backtracking is then to satisfy the new tighter trust-region constraint. In particular, if the model  $m_k$  is quadratic and if  $x_k + s_k$  minimizes it in the intersection of the trust-region  $\mathcal{B}_k$  and the line  $x_k + ts_k$  ( $t \geq 0$ ), then we are in the situation that we have just described, and the only action taken in the backtracking step is to scale down the step  $s_k$  to ensure that the new trial point lies in the smaller trust region.

We conclude this section by considering what can be said of the convergence properties of Algorithm 10.3.2. Theorem 8.1.10 (p. 267) ensures that when  $k - 1 \notin \mathcal{S}$ , the trial point  $x_k + s_k$  yields a sufficient decrease in the model, in that it satisfies AA.1b for the first-order criticality measure  $\pi(x) = \epsilon(x)\|\nabla_x f(x)\|$ . The same property obviously holds if  $k - 1 \in \mathcal{S}$  (in which case  $\epsilon(x) = 1$ ). We may thus apply the conclusions of Sections 8.1.2 and 8.1.3 and obtain that *all convergence properties of Chapter 6 again hold*. Observe in passing that we made sure, in AA.3 and Theorem 6.6.8 (p. 159), that the uniform bound on the trust-region radius may be imposed without interfering with this theory.

### 10.3.3 Which Steps Are Gradient Related?

Of course, we now have to investigate whether the gradient-relatedness condition imposed on  $s_k$  in Step 1a of Algorithm 10.3.2 is reasonable in that it is satisfied for known methods to compute  $s_k$ . Fortunately, this desirable property turns out to hold for a surprisingly large class of methods. Consider the decomposition of the step  $s_k$  into two parts given by

$$s_k = s_k^C + w_k, \quad (10.3.6)$$

where  $s_k^C$  is the step to the Cauchy point  $x_k^C$ . Because it is more appropriate here, we choose the generalized definition of the Cauchy point defined in Section 8.1.5, by (8.1.35) (p. 266) for a general gradient-related direction  $d_k$ . This decomposition is, of course, possible for any step  $s_k$ , since  $s_k^C = x_k^C - x_k$  is always well defined, and it suffices to set  $w_k = s_k - s_k^C$ . In the rest of this section, we will merely suppose that  $w_k$  is an ascent direction, that is, that

$$\langle g_k, w_k \rangle \leq 0. \quad (10.3.7)$$

This simple and very general property has the following important consequence.

**Theorem 10.3.2** Suppose that AF.1, AN.1, AM.3, and AM.4 hold, and that  $w_k$  is defined by (10.3.6) and satisfies (10.3.7). Suppose furthermore that there is a  $\Delta_{\max}$  such that

$$\Delta_k \leq \Delta_{\max}$$

for all  $k$ . Then  $s_k$  is gradient related, as defined in (8.1.48) (p. 268).

**Proof.** Suppose, without loss of generality, that the direction  $d_k$  satisfies the bound

$$\|s_k\| \leq \|d_k\| \leq \nu_k^C \Delta_k. \quad (10.3.8)$$

We have that

$$\begin{aligned} \langle g_k, s_k^C \rangle &= -t_k^C \langle g_k, d_k \rangle \\ &\leq -\kappa_{\text{dep}} \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\beta_k \|d_k\|}, \frac{\nu_k^C \Delta_k}{\|d_k\|} \right] \langle g_k, d_k \rangle \\ &\leq -\kappa_{\text{dep}} \epsilon(x_k) \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\beta_k \|d_k\|}, \frac{\nu_k^C \Delta_k}{\|d_k\|} \right] \|g_k\| \|d_k\|, \end{aligned}$$

where we have successively used the definition (8.1.35) (p. 266), Corollary 8.1.11 (p. 269), and (8.1.48) (p. 268). This, together with (10.3.6) and (10.3.7), gives that

$$\begin{aligned} |\langle g_k, s_k \rangle| &\geq \kappa_{\text{dep}} \|g_k\| \|s_k\| \epsilon(x_k) \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\beta_k \|d_k\|}, \frac{\nu_k^C \Delta_k}{\|d_k\|} \right] \\ &\geq \kappa_{\text{dep}} \|g_k\| \|s_k\| \epsilon(x_k) \min \left[ \frac{\epsilon(x_k) \|g_k\|}{\kappa_{\text{umh}} \kappa_{\text{une}} \Delta_{\max}}, 1 \right], \end{aligned} \quad (10.3.9)$$

where we have used AM.4, (10.3.8), and the bound  $\nu_k \Delta_k \leq \kappa_{\text{une}} \Delta_{\max}$  which results from AN.1. Using AM.3, we see that (10.3.9) is identical to (8.1.48) with  $\epsilon(x)$  replaced by

$$\epsilon'(x) = \kappa_{\text{dep}} \epsilon(x) \min \left[ 1, \frac{\epsilon(x) \|\nabla_x f(x)\|}{\kappa_{\text{umh}} \kappa_{\text{une}} \Delta_{\max}} \right].$$

This function is continuous because of AF.1, it has values in the interval  $[0, 1]$  because  $\kappa_{\text{dep}} \in (0, 1)$ , and it vanishes if and only if  $\nabla_x f(x) = 0$ . The step  $s_k$  is therefore gradient related.  $\square$

Because condition (10.3.7) is weak, this result covers a wide class of methods for computing the step  $s_k$ . In the special case where the model  $m_k$  is a quadratic, that is, of the form

$$m_k(x_k + s) = m_k(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle, \quad (10.3.10)$$

it covers, in particular, the Steihaug–Toint truncated conjugate gradient method (Algorithm 7.5.1), since this method first minimizes the model along the (preconditioned) steepest-descent direction, yielding the Cauchy point, and then pursues the minimization along further descent directions. This results from the following easy observation.

**Theorem 10.3.3** Suppose that  $s_k$  is computed using  $m$  iterations of Algorithm 7.5.1 (p. 205). Then,

$$\langle g_k, p_k^{(j)} \rangle \leq 0$$

for all  $j = 1, \dots, m$ , where  $p_k^{(j)}$  is the  $j$ th conjugate direction generated by this algorithm with  $M_k = I$ .

**Proof.** This immediately results from the relation  $\beta_k^{(i)} > 0$  ( $i = 0, \dots, m-1$ ) and the recurrence

$$\begin{aligned} \langle g_k, p_k^{(j)} \rangle &= \langle g_k^{(0)}, -g_k^{(j)} + \beta_k^{(j-1)} p_k^{(j-1)} \rangle \\ &= \beta_k^{(j-1)} \langle g_k, p_k^{(j-1)} \rangle \\ &= \prod_{i=0}^{j-1} \beta_k^{(i)} \langle g_k, p_k^{(0)} \rangle \\ &= -\|g_k\|^2 \prod_{i=0}^{j-1} \beta_k^{(i)}, \end{aligned}$$

where we have used the orthogonality of the gradients of the model,  $\langle g_k^{(0)}, g_k^{(j)} \rangle = 0$  ( $j \geq 1$ ), which is guaranteed by the conjugate gradient procedure (see (5.1.21) [p. 81]).  $\square$

Note that we have restricted ourselves to the case where the preconditioner  $M_k$  is the identity because we have considered the notion of gradient-relatedness (8.1.48) (p. 268) in the natural Euclidean inner product. Similar results can be derived if we

introduce a more general metric  $M_k$ , that is, if we consider the angle condition for the preconditioned gradient and direction.

Another important case is when  $x_k + s_k$  is the model minimizer  $x_k^M$ . The gradient-relatedness of  $s_k^M = x_k^M - x_k$  is the object of our next proposition.

**Theorem 10.3.4** Suppose that AF.1, AN.1, AM.3, and AM.4 hold. Suppose furthermore that  $m_k$  is a quadratic model of the form (10.3.10), and that there is a  $\Delta_{\max}$  such that

$$\Delta_k \leq \Delta_{\max}$$

for all  $k$ . Then  $s_k^M = x_k^M - x_k$  is gradient related, as defined in (8.1.48) (p. 268).

**Proof.** We know from Corollary 7.2.2 (p. 174) that there must exist a  $\lambda_k^M \geq 0$  such that

$$(H_k + \lambda_k^M I)s_k^M = -g_k, \quad \lambda_k^M(\|s_k^M\| - \nu_k^S \Delta_k) = 0, \quad (10.3.11)$$

and  $H_k + \lambda_k^M I$  is positive semidefinite. Consider first the case where  $\|s_k^M\| < \nu_k^S \Delta_k$ , which then implies that  $\lambda_k^M = 0$ . In this case we have that

$$\langle g_k, s_k^M \rangle = -\langle g_k, H_k^+ g_k \rangle \leq -\frac{\|g_k\|^2}{\lambda_{\max}[H_k]} \leq -\frac{\|g_k\|^2}{\beta_k},$$

which implies the desired result. Consider now the case where  $\|s_k^M\| = \nu_k^S \Delta_k$ . From (7.3.18) (p. 192), we know that

$$\lambda_{\min}[H_k] \leq -\lambda_k^M + \frac{\|g_k\|}{\nu_k^S \Delta_k}.$$

Using this inequality, the first part of (10.3.11), and the fact that

$$\lambda_{\max}[H_k] - \lambda_{\min}[H_k] \leq 2\|H_k\|,$$

we deduce that

$$\begin{aligned} \langle g_k, s_k^M \rangle &\leq -\langle g_k, (H_k + \lambda_k^M I)^+ g_k \rangle \\ &\leq -\frac{\|g_k\|^2}{\lambda_{\max}[H_k + \lambda_k^M I]} \\ &= -\frac{\|g_k\|^2}{\lambda_{\max}[H_k] + \lambda_k^M} \\ &\leq -\frac{\|g_k\|^2}{\lambda_{\max}[H_k] - \lambda_{\min}[H_k] + \frac{\|g_k\|}{\nu_k^S \Delta_k}} \\ &\leq -\frac{\|g_k\|^2}{2\|H_k\| + \frac{\|g_k\|}{\nu_k^S \Delta_k}} \\ &\leq -\frac{\|g_k\|^2}{2 \max \left[ \|H_k\|, \frac{\|g_k\|}{\nu_k^S \Delta_k} \right]} \end{aligned}$$

$$\leq -\frac{1}{2}\|g_k\| \min \left[ \nu_k^s \Delta_k, \frac{\|g_k\|}{2\beta_k} \right].$$

Thus we have that, in all cases,

$$\begin{aligned} |\langle g_k, s_k^M \rangle| &\geq \frac{1}{2}\|g_k\| \|s_k^M\| \min \left[ \frac{\|g_k\|}{2\beta_k \|s_k^M\|}, \frac{\nu_k^s \Delta_k}{\|s_k^M\|} \right] \\ &\geq \frac{1}{2}\|g_k\| \|s_k^M\| \min \left[ \frac{\|g_k\|}{2\kappa_{\text{umh}} \kappa_{\text{une}} \Delta_{\max}}, 1 \right], \end{aligned}$$

where we used our assumption on  $\Delta_k$ , AN.1, and the inequalities  $\|s_k^M\| \leq \nu_k \Delta_k$ ,  $\nu_k \leq \nu_{\max}$ , and AM.4. Now using AM.3, we observe that this bound is identical to (8.1.48) (p. 268) with

$$\epsilon(x) = \frac{1}{2} \min \left[ 1, \frac{\|\nabla_x f(x)\|}{2\kappa_{\text{umh}} \kappa_{\text{une}} \Delta_{\max}} \right].$$

This function is continuous because of AF.1, has values in the interval  $[0, 1]$ , is independent of  $s_k^M$ , and vanishes if and only if  $\nabla_x f(x) = 0$ . The step  $s_k^M$  is therefore gradient related.  $\square$

Note that we only used the right inequality of AN.1 in that we only required that  $\nu_k^s \Delta_k \leq \kappa_{\text{une}} \Delta_{\max}$ .

A third class of gradient-related steps is that described by the conditions (10.3.1) (see the proof of Theorem 10.3.1). Thus we may conclude that steps computed using the Steihaug–Toint truncated conjugate gradient method, or from the exact model minimizer, or for which conditions (10.3.1) are true may be used in Algorithm 10.3.2.

### Notes and References for Subsection 10.3.3

Combining trust-region and linesearch techniques is not a recent idea. Such a combination is used in the routines **VE08** (Toint, 1983b), and **VE10** (Toint, 1987b), for instance, where a linesearch is performed along the direction defined by the trust-region subproblem. Another contribution in this direction is that of Gertz (1999), where the linesearch process is additionally used to control the size of the trust-region radius. See also El-Hallabi (1999) for an application of the same idea in the context of constrained problems. The technique described in this section is due to Nocedal and Yuan (1998). They provided a more elaborate scheme where backtracking may be applied not only along the unsuccessful step  $s_k$ , but also along any sequence of directions  $\{d_k^{(i)}\}$  satisfying the conditions

$$\kappa_1 \|d_k^{(i)}\| \leq \|d_k^{(i+1)}\| \leq \kappa_2 \|d_k^{(i)}\|$$

for some  $0 < \kappa_1 < \kappa_2 < 1$ ,

$$\frac{\langle d_k^{(i)}, -g_k \rangle}{\|d_k^{(i)}\| \|g_k\|} \geq \frac{\langle s_k, -g_k \rangle}{\|s_k\| \|g_k\|},$$

and  $d_k^{(0)} = s_k$ . Their proposal is restricted to the case where quadratic models are used and  $s_k = s_k^M$ , possibly allowing for small inaccuracies in the computation of the model minimizer.

The assumption that  $\Delta_k \leq \Delta_{\max}$  is implicit in their development, as they suppose that the

iterates remain in a bounded set and the considered algorithm only increases  $\Delta_k$  if the iteration is very successful and the step  $s_k$  is at the boundary of the trust region. See also Yuan (1999). The extension to nonquadratic models and Theorem 10.3.2 appear to be new.

## 10.4 Other Measures of Model Accuracy

A question that arises in several useful contexts is how far we may modify the definition of the ratio  $\rho_k$  of achieved versus predicted reduction and still obtain the desirable convergence properties that we studied in Chapter 6. We will not consider this question in its full generality, but rather emphasize three cases of interest, based on our needs in later chapters. It will be immediately obvious that they have a common flavour, but we refrain from treating them in a more unifying framework because we feel that this approach would obscure the intuitive reasons why these modifications are proposed.

### 10.4.1 Magical Steps

We start by considering the case where, once a step  $s_k$  has been determined within the trust region, another *magical step*<sup>170</sup>  $s_k^{\text{MA}}$  is then given by some “oracle” such that the *composite step*  $s_k + s_k^{\text{MA}}$  is better than  $s_k$ , in the sense that

$$f(x_k + s_k + s_k^{\text{MA}}) \leq f(x_k + s_k). \quad (10.4.1)$$

Situations where such an oracle is available do occur in practice. For instance, it often happens that an expensive objective function contains a subset of variables that occurs in a predictable functional form (such as quadratically), and a second update can then be applied to those variables at very low cost after the original step  $s_k$  has been computed. A typical case is when the problem under consideration is in fact a subproblem in a more elaborate constrained minimization algorithm (see Section 14.4, for instance) and the objective function contains slack variables. In that case, the slack variables may be minimized, once  $x_k + s_k$  is known, to obtain (10.4.1) at very low cost. However, the resulting modified trust-region algorithm is not a special case of the basic algorithm, because condition (10.4.1) does not link the reduction obtained in the objective function to a predicted reduction in a model of this function. We thus have to reconsider the use of (10.4.1) and its associated convergence theory if we wish to understand the impact of the magical step  $s_k^{\text{MA}}$  in the context of our analysis of trust-region methods. Fortunately, this can be done quite simply, given the theory of Chapter 6.

We first restate an algorithm that is close to Algorithm BTR (p. 116), but includes the use of magical steps.

---

<sup>170</sup>In the linesearch context, the term *spacer step* is sometimes used instead of magical step. We prefer the latter because of the clearer dependence on some unspecified oracle, or, if you like, magic.

**Algorithm 10.4.1: Basic algorithm with magical steps**

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy conditions (6.1.3) (p. 116). Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m_k$  in  $\mathcal{B}_k$ .

**Step 2: Step calculation.** Compute a step  $s_k$  that “sufficiently reduces the model”  $m_k$  in the sense of AA.1 and such that  $x_k + s_k \in \mathcal{B}_k$ . Then compute a magical step  $s_k^{\text{MA}}$  such that (10.4.1) holds.

**Step 3: Acceptance of the trial point.** Define the ratio

$$\rho_k^{\text{MA}} = \frac{f(x_k) - f(x_k + s_k + s_k^{\text{MA}})}{m_k(x_k) - m_k(x_k + s_k) + f(x_k + s_k) - f(x_k + s_k + s_k^{\text{MA}})}. \quad (10.4.2)$$

If  $\rho_k^{\text{MA}} \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k + s_k^{\text{MA}}$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k^{\text{MA}} \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k^{\text{MA}} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k^{\text{MA}} < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

Note that only Steps 2 and 3 differ from those of Algorithm BTR, and this latter algorithm can be recovered by choosing  $s_k^{\text{MA}} = 0$ . Step 2 includes the calculation of the magical step  $s_k^{\text{MA}}$ , while Step 3 introduces a modified definition of the ratio  $\rho_k$  of predicted versus achieved reduction. The new ratio, denoted  $\rho_k^{\text{MA}}$ , again expresses the same idea<sup>171</sup> but now no longer measures the change from  $x_k$  to  $x_k + s_k$ , but from  $x_k$  to  $x_k + s_k + s_k^{\text{MA}}$ . However, since the improvement from  $x_k + s_k$  to  $x_k + s_k + s_k^{\text{MA}}$  is not based on any prediction using the model  $m_k$ , the denominator of  $\rho_k^{\text{MA}}$  now includes a contribution from the objective function itself, playing the role of its own model, as in Section 10.3.1. Note also that the algorithm requires the calculation of the objective function both at  $x_k + s_k$  and at  $x_k + s_k + s_k^{\text{MA}}$ . Finally, we observe, as we have already seen for the nonmonotone Algorithm 10.1.1, that Algorithm 10.4.1 may accept steps that would be rejected by Algorithm BTR. For we have that

$$\rho_k^{\text{MA}} \geq \rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

whenever  $\rho_k < 1$ , which is the case when the trial step is rejected in Algorithm BTR. The larger value of  $\rho_k^{\text{MA}}$  provides the possibility that this step could be accepted in Algorithm 10.4.1. This seems to be borne out in practice.

<sup>171</sup>The superscript in  $\rho_k^{\text{MA}}$  is a mnemonic for magical steps.

The adaptation of our convergence theory to cover Algorithm 10.4.1 is based on the observation that

$$\rho_k^{\text{MA}} = \frac{f(x_k) - f(x_k + s_k) + f(x_k + s_k) - f(x_k + s_k + s_k^{\text{MA}})}{m_k(x_k) - m_k(x_k + s_k) + f(x_k + s_k) - f(x_k + s_k + s_k^{\text{MA}})}.$$

This yields that

$$|\rho_k^{\text{MA}} - 1| = \frac{|\varepsilon_k|}{m_k(x_k) - m_k(x_k + s_k) + \alpha_k}, \quad (10.4.3)$$

where

$$\alpha_k = f(x_k + s_k) - f(x_k + s_k + s_k^{\text{MA}}) \geq 0$$

and

$$\varepsilon_k = m_k(x_k + s_k) - f(x_k + s_k). \quad (10.4.4)$$

Applying Theorem 6.4.1 (p. 133) to  $m_k$ , we deduce that

$$|\varepsilon_k| \leq \kappa_{\text{ubh}} \Delta_k^2. \quad (10.4.5)$$

On the other hand, taking AA.1 and AM.4 into account together with the bound  $\alpha_k \geq 0$  yields that

$$m_k(x_k) - m_k(x_k + s_k) + \alpha_k \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\kappa_{\text{ubh}}}, \Delta_k \right]. \quad (10.4.6)$$

Using this last bound, we may now derive the conclusion of Theorem 6.4.2 (p. 134).

**Theorem 10.4.1** Suppose that (10.4.5) and (10.4.6) hold and that  $m_k$  satisfies AM.1–AM.4. Suppose furthermore that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \|g_k\| (1 - \eta_2)}{\kappa_{\text{ubh}}}.$$

Then iteration  $k$  of Algorithm 10.4.1 is very successful and  $\Delta_{k+1} \geq \Delta_k$ .

**Proof.** The assumption on  $\Delta_k$  implies that

$$\Delta_k \leq \frac{\|g_k\|}{\kappa_{\text{ubh}}} \leq \frac{\|g_k\|}{\beta_k},$$

since  $\kappa_{\text{mdc}}(1 - \eta_2) \in (0, 1)$ . Hence we obtain that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \Delta_k \right] \geq \kappa_{\text{mdc}} \|g_k\| \Delta_k,$$

where we have used (10.4.6) and the bound  $\alpha_k \geq 0$  to derive the first inequality.

Combining this inequality with (10.4.5), we obtain that

$$|\rho_k^{\text{MA}} - 1| \leq \frac{\kappa_{\text{ubh}}}{\kappa_{\text{mdc}} \|g_k\|} \Delta_k \leq 1 - \eta_2.$$

Hence  $\rho_k^{\text{MA}} \geq \eta_2$  and iteration  $k$  is very successful. The mechanism of the algorithm then guarantees that  $\Delta_{k+1} \geq \Delta_k$ .  $\square$

The equivalents of Theorems 6.4.3 (p. 135) and 6.4.4 (p. 136) then immediately follow, because  $\rho_k$  does not appear at all in their proofs. The first global convergence result is then established as follows.

**Theorem 10.4.2** Suppose that AF.1, AF.2, AN.1, (10.4.5), AM.1–AM.4, and (10.4.6) hold. Suppose furthermore that Algorithm 10.4.1 is used. Then one has that

$$\liminf_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0. \quad (10.4.7)$$

**Proof.** The proof is identical to that of Theorem 6.4.5 (p. 136), except that (6.4.17) is replaced by

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1[m_k(x_k) - m_k(x_k + s_k) + \alpha_k] \\ &\geq \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_k \right], \end{aligned} \quad (10.4.8)$$

where we used (10.4.6) to derive the second inequality. This then implies that

$$f(x_k) - f(x_{k+1}) \geq \kappa_{\text{mdc}} \epsilon \eta_1 \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \kappa_{\text{lbd}} \right],$$

which is (6.4.17). The rest of the proof then follows without modification.  $\square$

Note that we could dispense with AF.1 and AM.3 if we only wished to show that

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0 \quad (10.4.9)$$

instead of (10.4.7). This latter result can therefore be obtained even for nonsmooth objective functions, provided (10.4.5) holds or at least that  $\varepsilon_k/\|s_k\|$  becomes arbitrarily small when  $\|s_k\|$  tends to zero. Also observe that, remarkably, Theorem 10.4.2 is true whether or not we have that  $\|s_k + s_k^{\text{MA}}\|_k \leq \Delta_k$ . However, we have to reintroduce a weak version of this condition if we want to prove that all limit points are first-order critical.

**Theorem 10.4.3** Suppose that AF.1–AF.2, AN.1, (10.4.5), AM.1–AM.4, and (10.4.6) hold. Suppose furthermore that Algorithm 10.4.1 is used and that, for all  $k$ ,

$$\|s_k^{\text{MA}}\| \leq \kappa_{\text{mms}} \Delta_k, \quad (10.4.10)$$

where  $\kappa_{\text{mms}}$  is a positive constant.<sup>172</sup> Then one has that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0.$$

---

<sup>172</sup>“mss” stands for “maximal step”.

**Proof.** The proof is identical to that of Theorem 6.4.6 (p. 137), except for the two following minor modifications.

The first is that (6.4.20) (p. 138) must now take into account that  $\rho_k^{\text{MA}}$  is used to determine which iterations are successful, instead of  $\rho_k$ . Hence, this equation may be rewritten as (10.4.8).

As in Theorem 6.4.6, we then deduce (6.4.21) (p. 138) and rewrite (6.4.22) (p. 138) as

$$\|x_{t_i} - x_{\ell_i}\| \leq \sum_{\substack{j=t_i \\ j \in \mathcal{K}}}^{\ell_i-1} \|x_j - x_{j+1}\| \leq \sum_{\substack{j=t_i \\ j \in \mathcal{K}}}^{\ell_i-1} (\nu_j^S + \kappa_{\text{mms}}) \Delta_j \leq \frac{\kappa_{\text{une}} + \kappa_{\text{mms}}}{\kappa_{\text{mde}} \epsilon \eta_1} [f(x_{t_i}) - f(x_{\ell_i})],$$

using AN.1 and the bound

$$\|s_k + s_k^{\text{MA}}\| \leq \|s_k\| + \|s_k^{\text{MA}}\| \leq \nu_k^S \Delta_k + \kappa_{\text{mms}} \Delta_k.$$

This last inequality is itself a consequence of the usual bound  $\|s_k\|_k \leq \Delta_k$  and (10.4.10). The proof may then be concluded as in Theorem 6.4.6.  $\square$

Again Lemma 6.5.1 (p. 140) and Theorem 6.5.2 (p. 141) extend in a straightforward way. Moreover, the attentive reader has probably already guessed why we may also generalize the proof of Lemma 6.5.3 (p. 144): the numerator of the fraction describing  $|\rho_k - 1|$  (or, in the present case,  $|\rho_k^{\text{MA}} - 1|$ ) only depends on  $\varepsilon_k$ , that is, on the adequacy of  $m_k$  as a model of  $f$ , which means that AM.5 and AF.1 are sufficient to obtain that  $|\varepsilon_k|/\|s_k\|_k^2$  is arbitrarily small when  $\|s_k\|$  is arbitrarily small. Combining this limit with (6.5.16) (p. 145) then allows us to finish the proof. The rest of the theory of Sections 6.5 and 6.6 also follows easily, using the fact that  $\alpha_k \geq 0$ , (10.4.3), (10.4.4), and (10.4.10), and replacing  $\kappa_{\text{une}}$  by  $\kappa_{\text{une}} + \kappa_{\text{mms}}$  whenever needed. As a consequence, we deduce that *all convergence properties of Algorithm BTR are preserved for Algorithm 10.4.1*. Obviously, this conclusion also generalizes to the framework of convergence for general criticality measures discussed in Section 8.1.

## 10.4.2 Correction Steps

We continue the line of thought started in the preceding section by examining our second case of interest. Its analysis is based on the observation that, except for deriving (10.4.3), we have not used the exact form of  $\alpha_k$  but have merely insisted that  $\alpha_k \geq 0$ . The question then naturally arises whether we could choose  $\alpha_k = 0$ , that is,

$$\rho_k^{\text{MA}} = \frac{f(x_k) - f(x_k + s_k + s_k^{\text{MA}})}{m_k(x_k) - m_k(x_k + s_k)}. \quad (10.4.11)$$

In this case, we have that

$$\varepsilon_k = f(x_k + s_k + s_k^{\text{MA}}) - m_k(x_k + s_k)$$

instead of (10.4.4). This means that this variant would be especially useful when  $f(x_k + s_k + s_k^{\text{MA}})$  is closer to  $m_k(x_k + s_k)$  than  $f(x_k + s_k)$ . Although this may seem unexpected at first glance, especially when AF.1 holds, such situations do nevertheless occur<sup>173</sup> when a nonsmooth objective function is involved. The step  $s_k^{\text{MA}}$  is then referred to as a *correction step* instead of a magical step. We may again revisit our convergence theory, and the above discussion indicates that no difficulties arise when proving convergence to first-order critical points, provided (10.4.5) and AA.1 hold. Similarly, the conclusion of Lemma 6.5.3 (p. 144) can be obtained in this context, provided that

$$\lim_{j \rightarrow \infty} \frac{|\varepsilon_{k_j}|}{\|s_{k_j}\|^2} = 0 \text{ for any subsequence } \{k_j\} \text{ such that } \lim_{j \rightarrow \infty} \|s_{k_j}\| = 0.$$

As a consequence, we may still derive the convergence results of Theorem 6.5.5 (p. 146) and of Section 6.6.3 if this latter condition can be shown to hold. Note that our comment on the fact that we haven't really used the differentiability properties of the objective function (that is AF.1 and AF.3) to obtain (10.4.9) is even more important in this context, which we anticipate to be most useful in the nonsmooth case.

### 10.4.3 Modified Models

The third case where modifying the definition of the ratio of achieved to predicted reduction is of interest is when the model  $m_k$  is naturally split into two components

$$m_k(x_k + s) = m_k^a(x_k + s) + \frac{1}{2}\langle s_k, Ms_k \rangle, \quad (10.4.12)$$

say, where  $m_k^a$  satisfies AM.1–AM.4, and where  $M_k$  is positive definite.<sup>174</sup> Thus  $m_k$  would itself satisfy AM.1–AM.4, the conditions that we require for proving convergence of the iterates generated by the basic algorithm to first-order critical points if, for all  $k$ ,

$$\|M_k\| \leq \kappa \quad (10.4.13)$$

for some  $\kappa > 0$ . Similarly,  $m_k$  would additionally satisfy AM.5 if  $m_k^a$  satisfies AM.5 and

$$\|M_k\| \rightarrow 0 \text{ when } \|g_k\| \rightarrow 0. \quad (10.4.14)$$

Second-order convergence results would then follow under AM.6, AA.2, and possibly AA.3. However, our purpose in this section is to show that (10.4.13) is not necessary when obtaining first-order convergence results; neither is (10.4.14) necessary for second-order convergence results, so long as one is ready to alter the definition of  $\rho_k$ .

Indeed, consider replacing  $\rho_k$  by

$$\rho_k^{\text{M}} = \frac{f(x_k) - f(x_k + s_k) - \frac{1}{2}\langle s_k, M_k s_k \rangle}{m_k(x_k) - m_k(x_k + s_k)} \quad (10.4.15)$$

---

<sup>173</sup>We will meet such cases in Chapter 11 and in Section 15.3.2.3.

<sup>174</sup>This situation occurs in Sections 13.12 and 13.12.2, for instance.

in Algorithm BTR or, equivalently, choosing  $s_k^{\text{MA}} = 0$  in Algorithm 10.4.1 with this new definition<sup>175</sup> of  $\rho_k^{\text{M}}$ . Then we immediately obtain that

$$|\rho_k^{\text{M}} - 1| = \frac{|\varepsilon_k|}{m_k(x_k) - m_k(x_k + s_k)}, \quad (10.4.16)$$

where we have defined

$$\varepsilon_k \stackrel{\text{def}}{=} f(x_k) - f(x_k + s_k) - m_k^a(x_k) + m_k^a(x_k + s_k) = m_k^a(x_k + s_k) - f(x_k + s_k), \quad (10.4.17)$$

and where we have used AM.2 for  $m_k^a$  to deduce the last equality. The result of this redefinition is that the (possibly unbounded) matrix  $M_k$  has disappeared from (10.4.16). Furthermore, (10.4.17) is identical to (10.4.4) in that  $\varepsilon_k$  is the difference between the values, at  $x_k + s_k$ , of the objective function and a model that satisfies AM.1–AM.4. As a consequence, we deduce from Theorem 6.4.1 (p. 133) that

$$|\varepsilon_k| \leq \kappa_{\text{uhb}} \Delta_k^2,$$

which is (10.4.5). On the other hand, AA.1 applies to  $m_k$  (and not to  $m_k^a$ ). As a consequence, we see that (10.4.6) again holds. As was the case for magical steps, we may therefore derive the results of Theorems 10.4.1, (6.4.13) (p. 135), and (6.4.15) (p. 136), which ensures that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k)\| = 0$$

and that every limit point (if any) is first-order critical. Thus we reach the conclusion that we may augment the model  $m_k^a$  by a (possibly unbounded) positive definite quadratic form without altering the first-order convergence properties of Algorithm BTR, provided we compensate by adapting the ratio of achieved versus predicted reduction according to (10.4.15). In effect, we have simply added the same term to both reductions, and this does not prevent their ratio from converging to one when such a limit would be obtained with the unmodified model. We note at this point that this technique somewhat blurs the intuitive interpretation of the model. Indeed, it is no longer clear what function the complete model (10.4.12) is modelling.

As could be anticipated, we may also dispense with (10.4.14) in the proof of Lemma 6.5.3 (p. 144): the argument is the same as that developed in the previous section for the case of magical steps. Hence, we deduce that *all convergence properties of Algorithm BTR<sup>176</sup> are preserved if we augment the model as in (10.4.12) but compensate by modifying  $\rho_k$  according to (10.4.15).*

## Notes and References for Section 10.4

Trust-region methods with magical steps are considered by Conn, Vicente, and Visweswaran (1999), who also examined the effects of such steps in the linesearch context. They used the

<sup>175</sup>The superscript in  $\rho_k^{\text{M}}$  is a mnemonic for “modified model”.

<sup>176</sup>Or of its extension to the general framework of convergence for general criticality measures discussed in Section 8.1.

condition (10.4.1) and the more complex definition (10.4.2), although we have seen that the simpler (10.4.11) achieves the same result. In addition, they showed that condition (10.4.10) in Theorem 10.4.3 can be replaced by

$$f(x_k + s_k) - f(x_k + s_k + s_k^{\text{MA}}) \geq \kappa \|s_k^{\text{MA}}\| \quad (10.4.18)$$

for some constant  $\kappa > 0$ . They report very encouraging results obtained with a modified version of the LANCELOT package. In their experiments, the magical steps result from a further minimization of the slack variables, in the context of problems with inequality constraints and nonlinear programming reformulations of minimax problems. They do not consider the (easy) extension of results on the convergence to second-order critical points when (10.4.10) is assumed. Note that it is unlikely that all these results hold if (10.4.10) is replaced by (10.4.18) (see the relation between  $\|x_{k+1} - x_k\|$  and  $\Delta_k$  in Theorems 6.6.7 and 6.6.8 [pp. 157, 159]).

Modifying the ratio  $\rho_k$  in order to take correction steps was first proposed by Fletcher (1982a, 1982b) and Yuan (1985b) in a method for minimizing nonsmooth objective functions, where they wished to take the computation of a “second-order correction step” into account. We will return to this problem in Section 15.3.2.3.

Finally, the form of modified models we have considered in Section 10.4.3 was proposed by Coleman and Li (1996b) in the context of the algorithm discussed in Section 13.12 for bound-constrained problems.

## 10.5 Alternative Trust-Region Management

### 10.5.1 Internal Doubling

The idea behind AA.3, that increasing the trust-region radius is beneficial when good agreement is obtained between the model and the objective function, leads to a further algorithmic extension of Algorithm BTR, sometimes called *internal doubling*. If we are given an initial trial radius at iteration  $k$  and if the preceding iteration was very successful, we may actually consider computing the step  $s_k$  for a sequence of successively larger radii,<sup>177</sup> so long as the successive steps increase in length and the successive values of  $\rho_k$  remain no smaller than  $\eta_1$ . The resulting algorithm<sup>178</sup> is Algorithm 10.5.1.

#### Algorithm 10.5.1: Basic algorithm with internal doubling

**Step 0: Initialization.** An initial point  $x_0$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1, \eta_2, \gamma_1, \gamma_2$ , and  $\gamma_3$  satisfy the conditions (6.1.3) (p. 116), and  $\gamma_3 > 1$ . Compute  $f(x_0)$  and set  $k = 0$  and  $\rho_{-1} = 1$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m_k$  in  $\mathcal{B}_k$ .

<sup>177</sup>The name “internal doubling” derives from the strategy of simply doubling the radius at each of these successive increases.

<sup>178</sup>Again, we ignore the possibility that  $m_k(x_k + s_k) = m_k(x_k)$ .

**Step 2: Initial step calculation.** Compute a step  $s_k$  that “sufficiently reduces the model”  $m_k$  and such that  $x_k + s_k \in \mathcal{B}_k$ . Compute  $f(x_k + s_k)$  and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

**Step 3: Inner iteration.** If  $\rho_{k-1} \geq \eta_2$  and  $\rho_k \geq \eta_2$ , set  $j = 0$ ,  $\Delta_k^{(0)} = \Delta_k$ .

**Step 3a: Enlarge the trust region.** Increment  $j$  by 1, choose  $\Delta_k^{(j)} \geq \gamma_3 \Delta_k^{(j-1)}$ , and define  $\mathcal{B}_k^{(j)} = \{x \in \mathbb{R}^n \mid \|x - x_k\|_k \leq \Delta_k^{(j)}\}$ .

**Step 3b: Inner step calculation.** Compute a step  $s_k^{(j)}$  that “sufficiently reduces the model”  $m_k$  and such that  $x_k + s_k^{(j)} \in \mathcal{B}_k^{(j)}$ . Compute  $f(x_k + s_k^{(j)})$  and define

$$\rho_k^{(j)} = \frac{f(x_k) - f(x_k + s_k^{(j)})}{m_k(x_k) - m_k(x_k + s_k^{(j)})}.$$

**Step 3c: Store the best step.** If  $\rho_k^{(j)} \geq \eta_1$  redefine

$$s_k = s_k^{(j)}, \quad \rho_k = \rho_k^{(j)}, \quad \Delta_k = \Delta_k^{(j)}.$$

If  $\rho_k^{(j)} \geq \eta_2$  and  $\|s_k^{(j)}\| > \|s_k^{(j-1)}\|$ , go to Step 3a.

**Step 4: Acceptance of the trial point.** If  $\rho_k \geq \eta_1$ , then set  $x_{k+1} = x_k + s_k$ ; otherwise set  $x_{k+1} = x_k$ .

**Step 5: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k) & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

The requirement that the previous iteration was very successful is made here for simplicity since it allows us to use the theory developed for Algorithm BTR. It may be justified by the desire to avoid reconsidering, as a result of these successive increases, a radius for which no successful step has been found before or a radius that was previously judged too large. Observe that in effect internal doubling introduces an *inner iteration* inside the main iteration of Algorithm BTR.

There are many possible variants of this algorithm. For instance, we may decide to limit the number of inner iterations to a (typically small) maximum number. We might also wish to use the test  $\rho_k \geq \eta_3$  for some  $\eta_3 \geq \eta_1$  to allow for inner iterations at the beginning of Step 3, together with the test  $\rho_k^{(j)} \geq \eta_3$  in Step 3c. Or we may decide to use, among the steps  $s_k$  accepted at Step 3c, that which produces the largest

decrease in the objective function, instead of the last one. Finally, we may allow for inner iterations only on selected iterations, such as the first.<sup>179</sup>

Internal doubling is clearly covered by the theory developed in Chapter 6, because any increase in the radius is admissible at very successful iterations of Algorithm BTR. This is because the final  $\Delta_k$  produced by the inner iteration at iteration  $k$  may be viewed as resulting from a very elaborate update to  $\Delta_k$  at the very successful iteration  $k - 1$ .

### Notes and References for Subsection 10.5.1

The first published mention of internal doubling appears to be on p. 145 in Dennis and Schnabel (1983). The version of this technique presented here is simple because it exploits the available theory for Algorithm BTR, but other variants that are not covered by this theory are also possible. For instance, one may remove the test that  $\rho_{k-1} \geq \eta_2$  at the beginning of Step 3 of Algorithm 10.5.1. Convergence theory covering this case may be found in Shultz, Schnabel, and Byrd (1985), where a more complicated model of the basic trust-region algorithm is analysed. An idea similar to internal doubling is discussed in Dennis et al. (1991) and Jarre (1998), where a curvilinear search is performed along the set of solutions of the  $\ell_2$  trust-region subproblem for varying radii. El-Hallabi (1999) investigates applying internal doubling in an algorithm for equality-constrained problems.

The observation that any increase, including no increase at all, is admissible at very successful iterations also allows for other algorithmic variants. For instance, Dennis and Mei (1979) consider quadratic models of the form (6.1.7) (p. 117) and only increase (double) the trust-region radius at very successful iterations if either

$$\|\nabla_x f(x_k + s_k) - \nabla_x f(x_k) - H_k s_k\| \leq \frac{1}{2} \|\nabla_x f(x_k)\| \quad \text{or} \quad \langle \nabla_x f(x_k), s_k \rangle \geq 2 \langle \nabla_x f(x_k + s_k), s_k \rangle.$$

The first of these conditions measures the quality of the model by the extent to which the Hessian matrix  $H_k$  is an adequate predictor of the change in gradient along the step, and it always holds if  $f(x)$  is itself quadratic and  $H_k = \nabla_{xx} f(x_k)$ . The second requires the slope of  $f$  along  $s_k$  to remain sufficiently negative, in which case one may hope for a better decrease in the objective function if larger steps are allowed.

### 10.5.2 Basing the Radius Update on the Steplength

If the norm of the current step  $\|s_k\|_k$  is much smaller than  $\Delta_k$ , one may argue that the ratio  $\rho_k$  does not measure the quality of the model in a ball of radius  $\Delta_k$ , but rather in a ball of radius  $\|s_k\|_k$ . Hence the update (6.1.5) (p. 116) may be viewed as inappropriate. In particular, if  $\rho_k$  is very small ( $< \eta_1$ ) or negative, it may take several iterations to reduce  $\Delta_k$  enough to ensure that  $x_k + s_k$  lies outside the trust region. Since each of these iterations requires, in theory, the evaluation of the objective function at the trial point, these evaluations would then be wasted. Fortunately, a simple modification of (6.1.5) prevents this undesirable situation without giving up any of the good properties

---

<sup>179</sup>This provides a way of diminishing the possibly negative influence of an initially too small  $\Delta_0$ .

of Algorithm BTR (p. 116): we simply have to replace the condition

$$\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k] \text{ when } \rho_k < \eta_1 \quad (10.5.1)$$

in (6.1.5) by

$$\Delta_{k+1} \in [\bar{\gamma}_1 \|s_k\|_k, \bar{\gamma}_2 \|s_k\|_k] \text{ when } \rho_k < \eta_1 \quad (10.5.2)$$

for some  $0 < \bar{\gamma}_1 < \bar{\gamma}_2 < 1$ . This ensures that  $s_k$  will no longer be acceptable at iteration  $k + 1$ , whenever

$$\|s_k\|_{k+1} \geq \|s_k\|_k. \quad (10.5.3)$$

Thus the resulting variant of the basic algorithm is more efficient in that a new step must be computed after at most one trust-region radius contraction. Note that (10.5.3) is not very restrictive, as in many cases the choice of the norm  $\|\cdot\|_k$  depends on  $x_k$  alone, and that  $x_{k+1} = x_k$  when  $\rho_k \leq \eta_1$ .

We now wish to verify that, under some additional (weak) conditions, the new variant is actually a special case of the basic algorithm and thus shares all its attractive convergence properties. Suppose, therefore, that the basic algorithm is applied instead of the above variant, and that iteration  $k$  is unsuccessful, as it must be because of the inequality of (10.5.1). Define  $m_{k+1} = m_k$  and  $\|\cdot\|_{k+1} = \|\cdot\|_k$ . By construction, the trust-region radius must be decreased at iteration  $k$ , and we shall achieve this by setting  $\Delta_{k+1} = \gamma_2 \Delta_k$ . If  $\|s_k\|_k \leq \Delta_{k+1}$ , then  $s_{k+1} = s_k$  is an acceptable step (in that it satisfies AA.1 and lies within the trust region  $\mathcal{B}_{k+1}$ ) and the new trial point is  $x_{k+1} + s_{k+1} = x_k + s_k$ . But, since we already know the objective function value at this point, there is no need to recompute  $f(x_{k+1} + s_{k+1})$ . By definition,  $\rho_{k+1} = \rho_k$  and iteration  $k + 1$  is also unsuccessful. Hence we have to reduce the trust-region radius, and we might again choose to reduce it by a factor  $\gamma_2$ . As before, we may define  $m_{k+2} = m_{k+1} = m_k$ ,  $\|\cdot\|_{k+2} = \|\cdot\|_{k+1} = \|\cdot\|_k$ , and  $s_{k+2} = s_{k+1} = s_k$ . This procedure can be continued for iterations  $k + j$ , choosing  $m_{k+j} = m_k$ ,  $\|\cdot\|_{k+j} = \|\cdot\|_k$ , and  $s_{k+j} = s_k$ , as long as

$$\|s_k\|_k \leq \gamma_2^j \Delta_k. \quad (10.5.4)$$

When this inequality eventually fails for some (smallest)  $j$  (as it must because  $\gamma_2 < 1$ ), we simply have to choose

$$\Delta_{k+j} \in [\gamma_1 \Delta_{k+j-1}, \gamma_2 \Delta_{k+j-1}] = [\gamma_1 \gamma_2^{j-1} \Delta_k, \gamma_2^j \Delta_k].$$

It is then not difficult to verify that (10.5.2) guarantees that this is the case, so long as

$$\gamma_1 \leq \gamma_2^2 \text{ and } \frac{\gamma_1}{\gamma_2} \leq \bar{\gamma}_1 < \bar{\gamma}_2 \leq \gamma_2. \quad (10.5.5)$$

Indeed, because  $j$  is the smallest integer for which (10.5.4) fails, we have that

$$\|s_k\|_k \leq \gamma_2^{j-1} \Delta_k \text{ and } \gamma_2^j \Delta_k < \|s_k\|_k$$

and therefore that

$$\left[ \frac{\gamma_1}{\gamma_2} \|s_k\|_k, \gamma_2 \|s_k\|_k \right] \subseteq [\gamma_1 \gamma_2^{j-1} \Delta_k, \gamma_2^j \Delta_k],$$

as desired. The variant involving (10.5.2) is thus a special case of Algorithm BTR if (10.5.3) and (10.5.5) hold.

The form of (10.5.2) and the discussion at the beginning of this section might then encourage us to completely replace the old trust-region radius  $\Delta_k$  by the scaled steplength  $\|s_k\|_k$  in the mechanism of Step 4 of Algorithm BTR, giving an update mechanism of the form

$$\Delta_{k+1} \in \begin{cases} [\|s_k\|_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \|s_k\|_k, \|s_k\|_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \|s_k\|_k, \gamma_2 \|s_k\|_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (10.5.6)$$

We have just verified that the third part of this condition is adequate. The attentive reader will also notice that there is no harm in using the second part either, as the convergence behaviour of trust-region methods is entirely governed by the first and third parts. However, the first part of (10.5.6) cannot be relied upon without further restrictions on the step calculation, as we now show by an example.

Consider the simple problem of minimizing the univariate function  $x^2$  using Algorithm BTR, where we make the choices  $\|\cdot\|_k = \|\cdot\|$  and  $m_k(x) = x^2$  for all iterations. Thus every iteration is very successful. Suppose now that  $x_0 > 0$  and that

$$s_k = \begin{cases} -\frac{1}{2} \operatorname{sgn}(x_k) \Delta_k & \text{if } |x_k| > \Delta_k, \\ -x_k & \text{otherwise} \end{cases}$$

at each iteration. From the convexity of the objective function, we deduce that

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}[m_k(x_k) - m_k(x_k^C)],$$

since we clearly have that

$$x_k^C = x_k^M = \begin{cases} x_k - \operatorname{sgn}(x_k) \Delta_k & \text{if } |x_k| > \Delta_k, \\ 0 & \text{otherwise.} \end{cases}$$

Thus AA.1 holds, and this should be enough to guarantee convergence, as all other assumptions are also satisfied. However, if we choose to define

$$\Delta_{k+1} = \|s_k\|_k = \frac{1}{2}\Delta_k,$$

as permitted by the first part of (10.5.6), we immediately see that

$$\Delta_k = \frac{1}{2^k} \Delta_0$$

and therefore that

$$x_k = x_0 + \sum_{i=1}^k s_i > x_0 - \Delta_0 \sum_{i=1}^{\infty} \frac{1}{2^i} = x_0 - \Delta_0.$$

Hence convergence to the unique first-order critical point (the origin) will not occur if  $x_0 > \Delta_0$ .

The problem is that the trust-region radius keeps on decreasing, although the iterations are very successful and the radius itself very small. This could be prevented by assuming, for instance, that

$$\|s_k\|_k \geq \|s_k^C\|_k, \quad (10.5.7)$$

where  $s_k^C$  is the step to the Cauchy point, because we have seen in the proofs of Theorems 6.3.1 (p. 125) and 6.3.3 (p. 128) that the length of this step is exactly  $\nu_k \Delta_k$  for small enough  $\Delta_k$ , so long as  $g_k$  remains bounded away from zero and  $\beta_k$  is bounded above (by  $\kappa_{ubh}$ ). Alternatively, we could directly require that

$$\|s_k\|_k = \Delta_k \text{ whenever } \|s_k^C\|_k = \Delta_k. \quad (10.5.8)$$

We leave it to the reader to verify that (10.5.7) or (10.5.8) are enough to ensure convergence if (10.5.6) is to be used. Note that Theorem 7.5.1 (p. 203) ensures that (10.5.7) holds if the step is computed by the Steihaug–Toint and generalized Lanczos trust-region methods (respectively, Algorithms 7.5.1 [p. 205] and 7.5.2 [p. 228]). Suppose now that  $s_k = s_k^M$ , the step to the global minimizer of a quadratic model within the trust region. If (10.5.7) is violated,  $x_k^M = x_k + s_k^M$  must belong to the interior of the trust region, which implies that the model is convex and that  $x_k^M$  may be found by running the Steihaug–Toint algorithm to termination. Theorem 7.5.1 then again guarantees that (10.5.7) must hold, which is impossible. Hence this condition cannot be violated in this case.

However, there does not seem to be any compelling reason to use the first part of (10.5.6) in practice, at least if  $\|s_k\|_k < \Delta_k$ . Reducing the trust-region radius when the model fits (very) well is somewhat counterintuitive and, more importantly, makes the algorithm less ambitious in its step determination. For this reason, we will not discuss this possibility any further. By contrast, we note that (10.5.2) is clearly useful because a good mechanism for reducing the radius at unsuccessful iterations is crucial for effective numerical performance.

### Notes and References for Subsection 10.5.2

Trust-region algorithms in which the trust-region radius is updated as a function of  $\|s_k\|$  instead of  $\Delta_k$  have been considered by many authors. For instance, methods using this approach have been proposed by Powell (1970d, 1970a), Thomas (1975), El-Hallabi (1993), El-Hallabi and Tapia (1993), Yuan (1993, 1998b), El-Alem (1995b), Dennis and Torczon (1997), and Dennis, Heinkenschloss, and Vicente (1998). In Gertz (1999), the idea is combined with the use of a linesearch along the direction resulting from the trust-region subproblem. We also refer the reader to our discussion in Section 17.1.

## 10.6 Problems with Dynamic Accuracy

We conclude this chapter on algorithmic extensions of the basic algorithm by considering the case where the objective function evaluation is the result of a computation

whose accuracy must be specified in advance. For instance, the evaluation of the objective may involve the solution of a nonlinear equation or be obtained after some integration procedure is applied. These calculations are themselves carried out with a certain accuracy, which has to be known before the evaluation. One may therefore wonder if an efficient algorithm could not exploit this feature by asking for sufficient accuracy in the value of the objective to guarantee progress of the minimization, while also relating the required accuracy to the level of noise on the objective function, with the ultimate goal of saving computing time for the evaluation itself.

In order to examine this possibility formally, we now suppose that the evaluation of the objective function at a vector  $x$  is no longer described by the simple value  $f(x)$ , but rather by some value  $\bar{f}(x, \epsilon)$ , where

$$|\bar{f}(x, \epsilon) - \bar{f}(x, 0)| \leq \epsilon, \quad (10.6.1)$$

where  $\epsilon \geq 0$  is the accuracy parameter. Further, we assume that

$$\bar{f}(x, 0) = f(x),$$

that is, the value of the objective function computed with full accuracy. We will therefore modify Algorithm BTR so that it specifies  $\epsilon$  as large as possible each time it must evaluate the objective function, while still maintaining its desirable global convergence properties. This immediately raises another question: as Algorithm BTR (p. 116) requires not only function values but also gradients, what accuracy should be required on the gradient, if an accuracy of  $\epsilon$  is demanded for the function value? Fortunately, this question does not introduce an additional level of complexity and may be answered by using the material described earlier in Section 8.4. For now, we will simply rely on AM.3b. This supposes that approximate gradients may be computed along with approximate function values, but that the quality of both approximations can be controlled separately. This is, for instance, the case when the objective function is computed by solving some discretized infinite-dimensional problem with an adaptive mesh: it is not uncommon that the gradient must then be computed in a slightly different manner, which may require a different mesh. In more general cases, we may need to apply the techniques that we have discussed in Section 9.4 for derivative-free minimization, but we will not elaborate specifically on this issue.

### 10.6.1 An Algorithm Using Dynamic Accuracy

The main idea developed in this subsection is that we only need to compute the objective with sufficient accuracy to ensure that the achieved decrease in function values observed after a successful iteration is not merely an effect of the inaccuracies in these values, but a true decrease. Furthermore, we must be able to compare that decrease to the decrease predicted by the model in order to take the decision to move to the new point or to stay at the current iterate. As the reader can see from this discussion, we have to determine the accuracy of the objective function values to make the parameter

$\rho_k$ , the ratio of achieved to predicted reduction, meaningful at the current iteration. More precisely, suppose that  $\epsilon_k$  and  $\epsilon_k^+$  are two values of the accuracy parameter. If we require that

$$\max[\epsilon_k, \epsilon_k^+] \leq \eta_0[m_k(x_k) - m_k(x_k + s_k)] \quad (10.6.2)$$

for some  $\eta_0 < \frac{1}{2}\eta_1$ , then we deduce from (10.6.1) that

$$\bar{f}(x_k, \epsilon_k) - f(x_k) \leq |\bar{f}(x_k, \epsilon_k) - f(x_k)| \leq \epsilon_k \leq \eta_0[m_k(x_k) - m_k(x_k + s_k)]$$

and also that

$$\begin{aligned} \bar{f}(x_k + s_k, \epsilon_k^+) - f(x_k + s_k) &\leq |\bar{f}(x_k + s_k, \epsilon_k^+) - f(x_k + s_k)| \\ &\leq \epsilon_k^+ \\ &\leq \eta_0[m_k(x_k) - m_k(x_k + s_k)]. \end{aligned}$$

These two inequalities in turn yield that

$$\begin{aligned} \frac{[\bar{f}(x_k, \epsilon_k) - f(x_k)] - [\bar{f}(x_k + s_k, \epsilon_k^+) - f(x_k + s_k)]}{m_k(x_k) - m_k(x_k + s_k)} &\leq \frac{\epsilon_k + \epsilon_k^+}{m_k(x_k) - m_k(x_k + s_k)} \\ &\leq \frac{2\eta_0}{2\eta_0}. \end{aligned}$$

Therefore the bound

$$\begin{aligned} \rho_k &= \frac{\bar{f}(x_k, \epsilon_k) - \bar{f}(x_k + s_k, \epsilon_k^+)}{m_k(x_k) - m_k(x_k + s_k)} \\ &= \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} + \frac{[\bar{f}(x_k, \epsilon_k) - f(x_k)] - [\bar{f}(x_k + s_k, \epsilon_k^+) - f(x_k + s_k)]}{m_k(x_k) - m_k(x_k + s_k)} \\ &\geq \eta_1 \end{aligned}$$

implies that

$$\frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} \geq \eta_1 - 2\eta_0 \stackrel{\text{def}}{=} \bar{\eta}_1 > 0.$$

This is crucial because it implies that condition (10.6.2) guarantees that a true decrease in the exact value of the objective function is obtained whenever  $\rho_k \geq \eta_1$  and that this decrease is at least a fraction  $\bar{\eta}_1$  of that predicted by the model. Conversely, if (10.6.2) does not hold, then we see that the reduction predicted by the model is not larger than the uncertainty of the function value, which indicates that the successful nature of the step  $s_k$  is very questionable. Note that (10.6.2) makes the accuracy requirement on the approximate values of  $f(x_k)$  and  $f(x_k + s_k)$  proportional to the predicted model decrease. Again, this is not unexpected: if the predicted decrease is large, a moderate uncertainty on these values is indeed tolerable. However, we may not simply add (10.6.2) to the statement of Algorithm BTR and obtain a new method that handles dynamic accuracy requirements on the objective function. The reason is that (10.6.2) requires in particular that

$$|\bar{f}(x_k, \epsilon_k) - f(x_k)| \leq \eta_0[m_k(x_k) - m_k(x_k + s_k)].$$

But the value of  $m_k(x_k + s_k)$ , which is produced within iteration  $k$ , is still unknown when the approximate value of  $f(x_k)$  is computed at iteration  $k - 1$ . This implies that

this approximation may have to be refined if its accuracy, specified at iteration  $k - 1$ , turns out to be insufficient in view of the model reduction predicted at iteration  $k$ . The resulting variant of Algorithm BTR is Algorithm 10.6.1.

**Algorithm 10.6.1: Algorithm with dynamic accuracy**

**Step 0: Initialization.** An initial point  $x_0$ , an initial trust-region radius  $\Delta_0$ , and an initial accuracy level  $\epsilon_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116), together with an  $\eta_0 \in (0, \frac{1}{2}\eta_1)$ . Compute  $f_0 = \bar{f}(x_0, \epsilon_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m_k$  in  $\mathcal{B}_k$  that satisfies AM.1,  $m_k(x_k) = f_k$ , AM.3b, and AM.4.

**Step 2: Step calculation.** Compute a step  $s_k$  that sufficiently reduces the model  $m_k$  (in the sense of AA.1) and such that  $x_k + s_k \in \mathcal{B}_k$ .

**Step 3: Verification of the objective's decrease accuracy.** If  $\epsilon_k$  does not satisfy (10.6.2), reduce the value of  $\epsilon_k$  to enforce this condition, recompute  $f_k = \bar{f}(x_k, \epsilon_k)$ , and return to Step 1. Otherwise, compute the accuracy level  $\epsilon_k^+$  satisfying (10.6.2) and calculate  $f^+ = \bar{f}(x_k + s_k, \epsilon_k^+)$ .

**Step 4: Acceptance of the trial point.** Define the ratio

$$\rho_k = \frac{f_k - f_k^+}{m_k(x_k) - m_k(x_k + s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$  and set  $\epsilon_{k+1} = \epsilon_k^+$ ; otherwise define  $x_{k+1} = x_k$  and set  $\epsilon_{k+1} = \epsilon_k$ .

**Step 5: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k) & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

We note that, although its existence has been assumed, the true value of the objective function at  $x_k$ ,  $f(x_k)$ , never appears in the algorithm, as intended. A first question immediately arises: is it possible that the loop between Steps 1 and 3 is infinite? In other words, can the modification of the predicted model reduction  $m_k(x_k) - m_k(x_k + s_k)$  result in a tightening of the resulting accuracy on the objective, itself resulting in a new model and a new predicted reduction, and so on? Fortunately, the answer is no: this situation cannot occur unless  $x_k$  is already a first- or second-

order critical point, depending on which assumptions are made on the computation of the step  $s_k$ .

**Lemma 10.6.1** Suppose that AM.3b, AM.4, and AA.1 hold. Then the loop between Steps 1 and 3 of Algorithm 10.6.1 is finite unless  $x_k$  is first-order critical. Furthermore, if AM.5, AM.6, and AA.2 also hold, this loop is finite unless  $x_k$  is second-order critical.

**Proof.** Suppose first that  $x_k$  is not first-order critical, that is,  $\nabla_x f(x_k) \neq 0$ . Lemma 8.4.1 (p. 281) then implies that there must exist a  $\delta > 0$  such that  $\|g_k\| \geq \delta$ . Applying this bound in AA.1 and using AM.4, we deduce that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min\left[\frac{\|g_k\|}{\beta_k}, \Delta_k\right] \geq \kappa_{\text{mdc}} \delta \min\left[\frac{\delta}{\kappa_{\text{umh}}}, \Delta_k\right].$$

As a consequence, if

$$\max[\epsilon_k, \epsilon_k^+] \leq \kappa_{\text{mdc}} \eta_0 \delta \min\left[\frac{\delta}{\kappa_{\text{umh}}}, \Delta_k\right],$$

then (10.6.2) will automatically be satisfied and neither  $\epsilon_k$  nor  $\epsilon_k^+$  needs to be reduced further. This proves that Step 4 will then be executed, and the first part of the result follows.

Suppose now that  $x_k$  is first-order but not second-order critical, that is, that

$$\nabla_x f(x_k) = 0 \quad \text{and} \quad \lambda_{\min}[\nabla_{xx} f(x_k)] = \tau_k < 0, \quad (10.6.3)$$

and also that AM.5, AM.6, and AA.2 hold. Suppose further, for the purpose of obtaining a contradiction, that the loop between Steps 1 and 3 is infinite. Then, Lemma 8.4.1 (p. 281) and the first part of (10.6.3) together imply that  $g_k = 0$ , in which case AM.5 gives that  $\nabla_{xx} f(x_k) = \nabla_{xx} m_k(x_k)$ , and therefore that

$$\tau_k = \lambda_{\min}[\nabla_{xx} m_k(x_k)].$$

We then deduce from Theorem 6.6.2 (p. 151), AM.6, and AA.2 that

$$m_k(x_k) - m_k(x_k + s_k) \geq -\kappa_{\text{sod}} \tau_k \min[\tau_k^2, \Delta_k^2].$$

Thus, if

$$\max[\epsilon_k, \epsilon_k^+] \leq -\kappa_{\text{sod}} \eta_0 \delta \tau_k \min[\tau_k^2, \Delta_k^2],$$

(10.6.2) must be satisfied and Step 4 will then be executed. This is impossible because we have assumed that the loop between Steps 1 and 3 is infinite, which brings the desired contradiction. Hence this loop must be finite if (10.6.3) holds, and the lemma is proved.  $\square$

Algorithm 10.6.1 is therefore well defined. Furthermore, either it produces a first-order critical iterate  $x_k$ , in which case an infinite loop between Steps 1 and 3 will be used to verify this property, or it generates an infinite sequence of iterates and then satisfies all the assumptions that we have used to prove global convergence of Algorithm BTR with inexact gradients (in Section 8.4.2). In both cases, global convergence to first-order critical points follows. Similarly, convergence to second-order critical points is obtained if we are ready to extend our assumptions to include AM.5–AM.6 and AA.2–AA.3.

We illustrate a possible sequence of iterations of Algorithm 10.6.1 in Figure 10.6.1. In this figure, the values of the objective function and models are shown vertically, and the successive iterations horizontally. Each thick vertical line represents an evaluation of the objective function. The true (unknown) value  $f(x_k)$  is indicated by a thick horizontal mark, the computed value  $\bar{f}(x_k, \epsilon_k) = m_k(x_k)$  by a small circle, and the amplitude of the possible error around this value, as specified by  $\epsilon_k$  in (10.6.1), by the extent of the thick vertical line. The curve starting from the computed value shows the decrease of the model along the computed step  $s_k$ , whose value  $m_k(x_k) - m_k(x_k + s_k)$  is made visible by the thin vertical line at the right end of the curve. In order to simplify the graph, the length of the steps is assumed to be uniform for all iterations, and all iterations are supposed to be successful. The initial iteration ( $k = 0$ ) starts with a value of  $\bar{f}(x_0, \epsilon_0)$  above  $f(x_0)$ , and then computes a relatively large model decrease

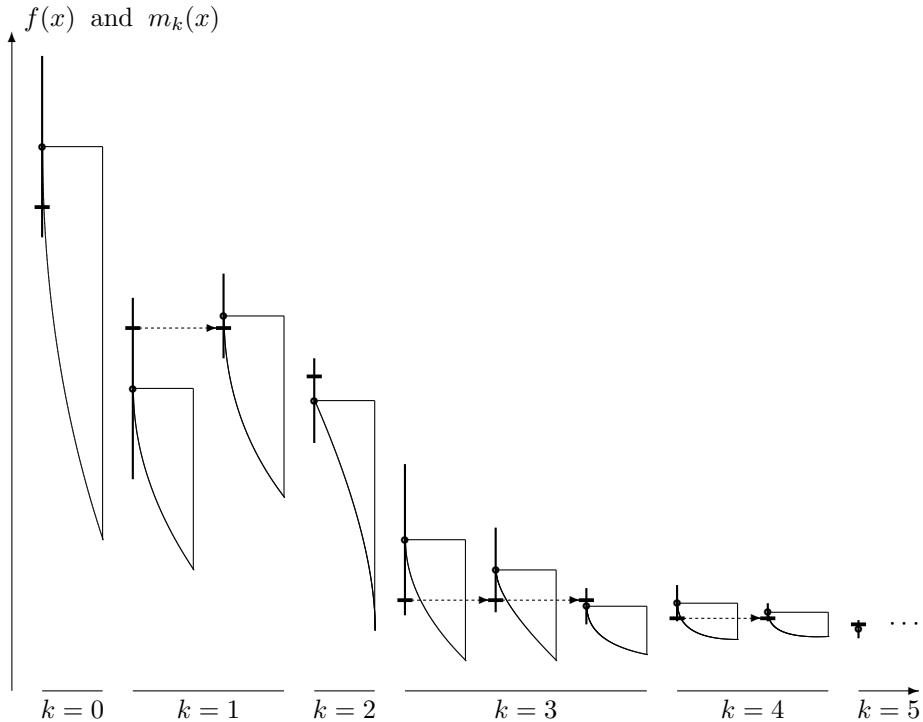


Figure 10.6.1: A few iterations of Algorithm 10.6.1.

$m_0(x_0) - m_0(x_0 + s_0)$ . This predicted decrease is large enough to satisfy

$$\epsilon_0 \leq \eta_0[m_0(x_0) - m_0(x_0 + s_0)]$$

but not large enough to choose  $\epsilon_0^+ > \epsilon_0$ : the value  $\bar{f}(x_0 + s_0, \epsilon_0^+)$  is therefore computed at Step 3 with  $\epsilon_0^+ = \epsilon_0$ , as shown on the next thick vertical line. Since this value is small enough for iteration 0 to be successful, we have that  $x_0 + s_0 = x_1$ ,  $\epsilon_1 = \epsilon_0^+ = \epsilon_0$ , and iteration 1 is started. At this iteration, the model decrease  $m_1(x_1) - m_1(x_1 + s_1)$  is too small for the inequality

$$\epsilon_1 \leq \eta_0[m_1(x_1) - m_1(x_1 + s_1)] \quad (10.6.4)$$

to hold. Hence  $\epsilon_1$  is reduced,  $\bar{f}(x_1, \epsilon_1)$  is recomputed (resulting in a value closer to  $f(x_1)$ ). Of course,  $f(x_1)$  has not changed, as indicated by the dashed arrow between the two successive evaluations of the objective at  $x_1$ . Then a new model and predicted decrease are calculated, which satisfies (10.6.4). The value of  $\bar{f}(x_1 + s_1, \epsilon_1^+)$  is then obtained (we have again chosen  $\epsilon_1^+ = \epsilon_1$ ) and iteration 2 successfully concluded. The next iterations follow the same general outline. We note that a negative curvature step is computed at iteration 2, resulting in a large predicted model decrease, which itself allows the choice  $\epsilon_2 = \epsilon_2^+ = \epsilon_3$ . The accuracy level  $\epsilon_3$  has to be reduced twice at iteration 3 in order to satisfy (10.6.2), and a similar reduction happens once at iteration 4. Observe that the sequence  $\{f(x_k)\}$  is decreasing but that this property does not hold for the sequence  $\{\bar{f}(x_k, \epsilon_k)\}$ .

We conclude this section with two practical considerations. The first is that the reduction of  $\epsilon_k$  and  $\epsilon_k^+$  in Step 3 does not necessarily lead to a large computational expense. If we suppose, as is common in this context, that  $\bar{f}(x, \epsilon)$  is computed by some convergent iterative process, we may simply restart this iteration from where it was previously stopped, in order to obtain better accuracy in the value of the objective function.

A second observation is that the framework we just discussed covers the interesting case where the evaluation of the objective function is subject to some random noise, with the property that the noise does not depend uniquely on  $x$ , that is, that two evaluations of the objective at  $x$  return two (potentially) different approximations of the true value  $f(x)$ . This could, for instance, be the case if  $f(x)$  involves some physical measurement, in which the same error is not typically repeated if the measure is taken again. Evaluating  $\bar{f}(x, \epsilon)$  for a given  $\epsilon$  then amounts to evaluating the noisy objective a number of times, which is sufficient to ensure that the average of these evaluations satisfies the bound (10.6.1) with some confidence level. For example, if we suppose that the noise on the evaluation at  $x$  follows a normal distribution of mean  $f(x)$  and variance  $\sigma(x)^2$ , then evaluating  $\bar{f}(x, \epsilon)$  with a 95% confidence level requires at least

$$\left[1.96 \frac{\sigma(x)}{\epsilon}\right]^2$$

evaluations of the noisy objective at  $x$ , as is well known from elementary statistics.<sup>180</sup> Again, reducing  $\epsilon$  does not imply that the whole process of computing  $\bar{f}(x, \epsilon)$  should be restarted, since the previous evaluations at  $x$  could be used to build the new and more accurate average. Note that one would typically use the framework of interpolation models developed in Section 9.4 in this latter case, instead of relying on AM.3b.

### 10.6.2 Conditional Models and Dynamic Accuracy

The framework that we have developed for handling dynamic accuracy levels in the evaluation of the objective function is not unconnected to that of conditional models studied in Section 9.1. In fact, condition (10.6.2) could be viewed as a condition for the model  $m_k$  to be valid, in that it ensures a sufficient accuracy of the model with respect to the true objective function. Conversely, it is natural to ask whether a condition like (10.6.2) could replace the validity condition (9.1.2) (p. 308) in the setting of conditional models. The main difficulty with this suggestion is that (10.6.2) only specifies the accuracy of the model at  $x_k$  and  $x_k + s_k$ , and not within some neighbourhood  $Q_k(\delta)$  of  $x_k$ . As a consequence, Theorem 9.1.1 (p. 308) and (9.3.1) (p. 319) no longer hold, and the accuracy of the first (and second) derivative(s) can no longer be estimated. This is exactly why we need AM.3b (and AM.5) in conjunction with (10.6.2).

Another difference between the two settings is that (9.1.2) can sometimes be proved (as in Theorem 9.4.4 [p. 333]) despite the fact that only a possibly loose upper bound on the precise value of constant  $\kappa_{\text{cnd}}$  is known. Condition (10.6.2), by contrast, leaves very little freedom of that nature, because of the condition  $\eta_0 < \frac{1}{2}\eta_1$ .

## Notes and References for Section 10.6

Our presentation of trust-region methods for noisy problems is based on Conn et al. (1993), but includes an explicit strategy to define the dynamic accuracy level, when it can be specified. The interest of a framework including separate control for the objective and gradient errors was pointed out to us by H. G. Bock in the context of multiple shooting methods for parameter identification. See, for instance, Bock, Schlöder, and Schulz (1995). Carter (1993) gave evidence that trust-region methods are relatively robust in the presence of noise on both objective function and gradient values. A derivative-free algorithm for noisy functions is also proposed by Elster and Neumaier (1997).

---

<sup>180</sup>See Wonnacott and Wonnacott (1990), p. 258, for example.

# Chapter 11

---

## Nonsmooth Problems

---

In this chapter, we turn to the case where  $f(x)$  is continuous but not necessarily smooth. In fact, we shall study a narrower class of problems, namely, those that are locally Lipschitz continuous and regular in the region of interest. We recall from Section 3.1.2 that a function  $f$  is locally Lipschitz continuous on  $\mathbb{R}^n$  if there are strictly positive values  $\gamma(x)$  and  $\epsilon(x)$  for which

$$|f(z) - f(x)| \leq \gamma(x)\|z - x\|$$

for all  $z \in \mathcal{O}_{\epsilon(x)}(x)$  and all  $x \in \mathbb{R}^n$ . It is regular (Section 3.1.4) if its one-sided directional derivative  $f'_d(x)$  exists for all  $x$  and  $d \in \mathbb{R}^n$  and the one-sided and generalized directional derivatives agree, that is, if

$$f'_d(x) = f_d^o(x).$$

While this is inevitably a restriction, it does cover a number of practically important classes of problems, namely, those that involve convex functions or those composed of convex and differentiable functions. In particular, if  $c$  is a continuously differentiable function from  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathcal{C} \subset \mathbb{R}^m$ , it allows us to consider the problems of minimizing  $\|c(x)\|$  or  $\|\max[c(x), 0]\|$  for any monotonic norm,<sup>181</sup> which includes all the  $\ell_p$  norms. Thus we are able to find the minimum-norm solution to a system of equations  $c(x) = 0$  (we will consider this problem in a variety of different  $\ell_p$  norms in Sections 16.1 and 16.2), or to find the least infeasible point to a system of inequalities  $c(x) \leq 0$ , and hence to find feasible points to such systems, or even to minimize composite functions like  $f(x) + \|\min[c(x), 0]\|$ , which has implications for constrained optimization (see Chapter 15). It also allows us to cover slightly more general nonconvex functions like

$$f(x) = \begin{cases} x^3 - 3x^2 + 2x + 1 & \text{if } x \geq 0, \\ -x^3 - 3x^2 - 2x + 1 & \text{if } x < 0, \end{cases} \quad (11.0.1)$$

which we illustrate in Figure 11.0.1. We shall give a number of important examples in Section 11.4. For future reference, we formally assume the following.

<sup>181</sup>A norm is monotonic if  $|x| \leq |y|$  implies that  $\|x\| \leq \|y\|$ . Not all norms are monotonic. For example, the  $H$  norm,  $\|x\|_H^2 = \langle x, Hx \rangle$ , is only monotonic if the positive definite matrix  $H$  is diagonal.

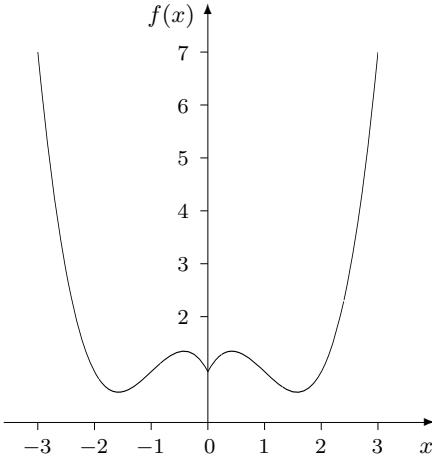


Figure 11.0.1: A plot of  $x$  against  $f(x)$  for the function (11.0.1). Notice that the function is globally nonconvex, but convex in a neighbourhood of its derivative discontinuity.

**AF.1n** The function  $f(x)$  is locally Lipschitz continuous and regular on  $\mathbb{R}^n$ .

We now consider algorithms for minimizing such functions.

## 11.1 Algorithms for Nonsmooth Optimization

Remarkably, the simple framework, Algorithm BTR (p. 116), that we introduced for smooth problems is still appropriate here, provided we replace the trust-region radius update (6.1.5) by

$$\Delta_{k+1} \in \begin{cases} [\gamma_3 \Delta_k, \Delta_{\max}] & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1, \end{cases}$$

where  $0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3$  (cf. AA.3, but note the extra requirement  $1/\gamma_3 \leq \gamma_1$ ), and so long as we make suitable redefinitions of what we now mean by the model, and what a “sufficient” reduction entails. The model is, as before, merely intended to be an easily manipulated simplification of  $f(x + s)$  that accurately represents  $f$  when  $s$  is small. For generality, we allow the model to depend on a vector of parameters,  $p \in \mathcal{P}$ , that may be adjusted at each iteration, and thus write  $m(x, p, s)$  for the model. Examples of such parameters might be approximate second derivatives, Lagrange multipliers, or penalty parameters, depending on the context. We shall require the following assumption.

**AM.1n** The model  $m(x, p, s)$  is locally Lipschitz continuous and regular with respect to  $s$  for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ , and continuous in  $(x, p)$  for all<sup>182</sup>  $s \in \mathbb{R}^n$ .

<sup>182</sup>Strictly, this is only required for  $s$  lying within an appropriate trust region around  $x$ .

**AM.2n** The values of the objective function and of the model coincide when  $s = 0$ ; that is,

$$m(x, p, 0) = f(x) \quad (11.1.1)$$

for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ .

**AM.3n** The generalized directional derivatives of the model and of the objective function coincide in every nonzero direction  $d$  when  $s = 0$ ; that is,

$$m_d^o(x, p, 0) = f_d^o(x) \text{ for all } d \neq 0 \in \mathbb{R}^n \quad (11.1.2)$$

for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ .

**AM.4n** The set of parameters  $\mathcal{P}$  is closed and bounded.

Assumption AM.1n imposes a slight additional restriction over what we have seen in previous chapters since it requires continuity with respect to  $p$ —previously, for example, the second-derivatives approximation might change on unsuccessful iterations, which would be a discontinuous change with  $k$ . Assumptions AM.2n and AM.3n simply require that the model be at least a first-order approximation of  $f$ . The biggest restriction here is AM.4n, since it is known that algorithms for specific nonsmooth problems do not require it (see the notes at the end of Section 11.5). We make this assumption purely so we can derive a general convergence theory.

We shall make much use of the steepest-descent direction  $d(x)$  at  $x$ , which we define to be

$$d(x) = \arg \min_{\|d\| \leq 1} f_d^o(x); \quad (11.1.3)$$

we denote the slope in this direction as

$$\pi(x) = \min_{\|d\| \leq 1} f_d^o(x).$$

We now show how the steepest-descent direction, and its slope, may be expressed purely in terms of the generalized gradient of  $f$  at  $x$ .

**Lemma 11.1.1** Suppose that

$$\pi(x) = \min_{\|d\| \leq 1} f_d^o(x).$$

Then

$$\pi(x) = -\|g(x)\|,$$

where

$$g(x) = \arg \min_{g \in \partial f(x)} \|g\|. \quad (11.1.4)$$

**Proof.** It follows directly from (3.1.4) (p. 33) that

$$\pi(x) = \min_{\|d\| \leq 1} f_d^o(x) = \min_{\|d\| \leq 1} \max_{g \in \partial f(x)} \langle g, d \rangle.$$

We also have that, for any  $d_*$  for which  $\|d_*\| \leq 1$  and  $g_* \in \partial f(x)$ ,

$$\min_{\|d\| \leq 1} \langle g_*, d \rangle \leq \langle g_*, d_* \rangle \leq \max_{g \in \partial f(x)} \langle g, d_* \rangle \quad (11.1.5)$$

since both sets are compact. Thus, picking  $d_*$  and  $g_*$  to maximize and minimize, respectively, the left- and right-hand sides of (11.1.5), and using the Cauchy–Schwarz inequality, it follows that

$$-\|g(x)\| = -\min_{g \in \partial f(x)} \|g\| = \max_{g \in \partial f(x)} (-\|g\|) \leq \max_{g \in \partial f(x)} \min_{\|d\| \leq 1} \langle g, d \rangle \leq \min_{\|d\| \leq 1} \max_{g \in \partial f(x)} \langle g, d \rangle.$$

It is then straightforward to show that this inequality is actually an equation. For, let

$$d(x) = \begin{cases} -g(x)/\|g(x)\| & \text{if } g(x) \neq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (11.1.6)$$

The result is immediate if  $g(x) = 0$ . So, suppose  $g(x) \neq 0$ . Then

$$\begin{aligned} -\|g(x)\| &\leq \min_{\|d\| \leq 1} \max_{g \in \partial f(x)} \langle g, d \rangle \\ &\leq \max_{g \in \partial f(x)} \langle g, d(x) \rangle \\ &= -\frac{\min_{g \in \partial f(x)} \langle g, g(x) \rangle}{\|g(x)\|} \\ &= -\frac{\min_{g \in \partial f(x)} \|g(x)\|^2 + \langle g - g(x), g(x) \rangle}{\|g(x)\|} \\ &= -\|g(x)\| - \frac{\min_{g \in \partial f(x)} \langle g - g(x), g(x) \rangle}{\|g(x)\|} \\ &\leq -\|g(x)\|, \end{aligned}$$

the last inequality following from the characterization (3.1.2) (p. 31) of  $g(x)$  as the solution of the least-distance problem (11.1.4).  $\square$

Notice that (11.1.6) is an explicit expression for the steepest-descent direction and that  $x$  is a first-order critical point if and only if  $\pi(x) = 0$ .

**Lemma 11.1.2** Suppose that AF.1n holds and that  $x_*$  is not a first-order critical point of  $f$ . Then there are strictly positive values  $\epsilon$  and  $\kappa_{\text{psg}}$  such that

$$\|g(x)\| \geq \kappa_{\text{psg}}$$

for all  $x \in \mathcal{O}_\epsilon(x_*)$ , where  $g(x)$  is defined as in (11.1.4).

**Proof.** Clarke's and Rademacher's theorems together show that there must be another point  $\partial f(x_*)$  that is the convex hull of all limit points of gradients of points converging to  $x_*$  at points where the gradient exists. Since  $f$  is locally Lipschitz, Rademacher's theorem (see Section 3.1.2) shows that  $f$  is differentiable almost everywhere. Therefore, there are points in every neighbourhood of  $x_*$  at which  $f$  is differentiable.

Let  $\{\epsilon_i\}$  and  $\{\delta_i\}$  be sequences of positive scalars converging to zero, and suppose that in any neighbourhood  $\mathcal{N}_i$  of  $x_*$ , there is a point  $x_i$  which is critical. For each  $x_i$ , Rademacher's theorem implies that there is another point  $y_i \in \mathcal{O}_{\epsilon_i}(x_i)$  for which  $\|\nabla_x f(y_i)\| \leq \delta_i$ . Therefore, since  $y_i$  converges to  $x_*$  as  $i$  increases,  $0 \in \partial f(x_*)$ , which contradicts our hypothesis that  $x_*$  is not critical. Thus, there are no critical points within some neighbourhood  $\mathcal{O}_\epsilon(x_*)$  of  $x_*$ .

Now suppose that in any neighbourhood  $\mathcal{N}_i$  of  $x_*$ , there is a point  $x_i$  for which  $\|g(x_i)\| \leq \delta_i$ . Clarke's and Rademacher's theorems then together show that there must be another point  $y_i \in \mathcal{O}_{\epsilon_i}(x_i)$  for which  $\|\nabla_x f(y_i)\| \leq 2\delta_i$ . Since  $y_i$  converges to  $x_*$  as  $i$  increases,  $0 \in \partial f(x_*)$ , which again contradicts our hypothesis, and establishes the lemma.  $\square$

The key ingredient in the development of a useful step, assumption AA.1 in Chapter 6, was to require that the decrease in the model was at least a fixed fraction of that achievable from the Cauchy step. The same will be true here. We define the Cauchy step for the model problem

$$\begin{aligned} & \text{minimize } m(x, p, s) \\ & \quad \begin{matrix} s \in \mathbb{R}^n \\ \|s\| \leq \Delta \end{matrix} \end{aligned} \tag{11.1.7}$$

to be

$$s^C(x) = t^C(x)d(x), \quad \text{where } t^C(x) = \arg \min_{0 < t \leq \Delta} m(x, p, td(x)) \tag{11.1.8}$$

and  $d(x)$  is the steepest-descent direction (11.1.3). We shall choose a step  $s(x)$  for which  $\|s(x)\| \leq \Delta$  and for which

$$m(x, p, 0) - m(x, p, s(x)) \geq \kappa_{\text{mdc}}(m(x, p, 0) - m(x, p, s^C(x))) \tag{11.1.9}$$

for some  $\kappa_{\text{mdc}} > 0$ . However, this formulation requires that we know the value of the Cauchy step, and we shall shortly pose a more general requirement that includes the Cauchy step as a particular case.

Since we have not assumed a quadratic model of the form (6.1.7) (p. 117), nor can we invoke a mean value theorem to approximate  $f$  by a quadratic, it is difficult, in general, to justify assuming a global "sufficient" decrease condition like (6.3.27) (p. 131). However, we can use the following local result.

**Lemma 11.1.3** Assume that AM.1n and AM.3n hold. Then given any  $x_*$ , there are values  $\delta > 0$  and  $\epsilon > 0$  such that<sup>183</sup>

$$m(x, p, 0) - m(x, p, \Delta d(x)) \geq \frac{1}{2} \|g(x)\| \Delta \quad (11.1.10)$$

for all  $\Delta \leq \delta$ ,  $x \in \mathcal{O}_\epsilon(x_*)$  and  $p \in \mathcal{P}$ .

**Proof.** Suppose otherwise that, for any sequences  $\{\delta_k\}$  and  $\{\epsilon_k\}$  of positive values whose limit is zero, there are values  $x_k$ ,  $p_k$ , and  $\Delta_k$  for which  $\|x_k - x_*\| \leq \epsilon_k$ ,  $\Delta_k \leq \delta_k$ , and

$$m(x_k, p_k, 0) - m(x_k, p_k, \Delta_k d(x_k)) < \frac{1}{2} \|g(x_k)\| \Delta_k$$

or, equivalently,

$$\frac{m(x_k, p_k, \Delta_k d(x_k)) - m(x_k, p_k, 0)}{\Delta_k} > -\frac{1}{2} \|g(x_k)\| \quad (11.1.11)$$

for all  $k$ . However, it follows from AM.1n, AM.3n, and Lemma 11.1.1 that

$$\begin{aligned} -\|g(x_k)\| &= f_{d(x_k)}^o(x_k) = m_{d(x_k)}^o(x_k, p_k, 0) \\ &= \lim_{\Delta_k \searrow 0} \frac{m(x_k, p_k, \Delta_k d(x_k)) - m(x_k, p_k, 0)}{\Delta_k} \end{aligned}$$

and thus that

$$\frac{m(x_k, p_k, \Delta_k d(x_k)) - m(x_k, p_k, 0)}{\Delta_k} \leq -\frac{3}{4} \|g(x_k)\| \quad (11.1.12)$$

for all  $\Delta_k$  sufficiently small. The result then follows as (11.1.12) contradicts (11.1.11).  $\square$

**Corollary 11.1.4** Assume that AM.1n and AM.3n hold. Then given any  $x_*$  and  $\Delta > 0$ , there are values  $\delta > 0$  and  $\epsilon > 0$  such that

$$m(x, p, 0) - m(x, p, s^C(x)) \geq \frac{1}{2} \|g(x)\| \min [\delta, \Delta] \quad (11.1.13)$$

for all  $x \in \mathcal{O}_\epsilon(x_*)$  and  $p \in \mathcal{P}$ .

**Proof.** In view of Lemma 11.1.3, there are two cases to consider. If  $\Delta < \delta$ , (11.1.10) implies that

$$m(x, p, 0) - m(x, p, s^C(x)) \geq m(x, p, 0) - m(x, p, \Delta d(x)) \geq \frac{1}{2} \|g(x)\| \Delta.$$

<sup>183</sup>The constant  $\frac{1}{2}$  here is not crucial and may be replaced by any value in  $(0, 1)$ .

On the other hand, if  $\Delta \geq \delta$ , (11.1.10) implies that

$$m(x, p, 0) - m(x, p, s^C(x)) \geq m(x, p, 0) - m(x, p, \delta s^C(x)) \geq \frac{1}{2} \|g(x)\| \delta.$$

Combining these two inequalities yields (11.1.13).  $\square$

In view of (11.1.9) and Corollary 11.1.4, we say that  $s$  is a *suitable* step at  $x$  if there exist  $\delta > 0$ ,  $\epsilon > 0$ , and a constant  $\kappa_{\text{mdc}} \in (0, 1)$  such that  $s$  satisfies  $\|s\| \leq \Delta$  and

$$m(x, p, 0) - m(x, p, s) \geq \kappa_{\text{mdc}} \|g(x)\| \min [\delta, \Delta] \quad (11.1.14)$$

for all  $x \in \mathcal{O}_\epsilon(x_*)$  and  $p \in \mathcal{P}$ . More formally, we assume the following.

**AA.1n** For any given  $x_*$ , there exist constants  $\delta, \epsilon > 0$  and  $\kappa_{\text{mdc}} \in (0, 1)$ , such that the step  $s_k$  satisfies

$$m(x_k, p_k, 0) - m(x_k, p_k, s_k) \geq \kappa_{\text{mdc}} \|g(x_k)\| \min [\delta, \Delta_k],$$

whenever  $x_k \in \mathcal{O}_\epsilon(x_*)$ .

Notice that, from a practical point of view, since Lemma 11.1.4 implies that (11.1.14) is satisfied by a step to the boundary in the steepest-descent direction, it will also be satisfied by any other step that produces a reduction in the model at least as large as that at the Cauchy point (11.1.8). This then is a natural generalization of assumption AA.1 in Chapter 6. We will not suggest, at this stage, how one might find such a step, or indeed even how we might compute the Cauchy step, but merely observe that both the Cauchy step and the global minimizer of the model give suitable steps.

We are now in a position to state our general nonsmooth trust-region algorithm, Algorithm 11.1.1.

### Algorithm 11.1.1: Basic nonsmooth trust-region algorithm

**Step 0: Initialization.** An initial point  $x_0$  and initial and maximum permissible trust-region radii  $0 < \Delta_0 \leq \Delta_{\max}$  are given. The constants  $\eta_1, \eta_2, \gamma_1, \gamma_2$ , and  $\gamma_3$  are also given and satisfy the conditions

$$0 < \eta_1 \leq \eta_2 < 1 \text{ and } 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3. \quad (11.1.15)$$

Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m(x_k, p_k, s)$  in  $\mathcal{B}_k$  satisfying AM.1n–AM.4n.

**Step 2: Step calculation.** Compute a step  $s_k$  satisfying AA.1n and for which  $x_k + s_k \in \mathcal{B}_k$ .

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m(x_k, p_k, 0) - m(x_k, p_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\gamma_3 \Delta_k, \Delta_{\max}] & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2], \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1, \end{cases} \quad (11.1.16)$$

Increment  $k$  by 1 and go to Step 1.

Typical values are exactly as for Algorithm BTR (p. 116), with additionally  $\gamma_3 = 2$  and  $\Delta_{\max} = 10^{10}$ , but as before, other values may be preferable.

## 11.2 Convergence to a First-Order Critical Point

We now consider the convergence of Algorithm 11.1.1. As usual, we need to prove a sequence of preliminary lemmas before we finally establish our global convergence result.

While assumption AM.3n may seem reasonable, it is not always easy to check, and we thus seek a more convenient but equivalent form. To this end, we have the following result.

**Lemma 11.2.1** Suppose assumptions AF.1n and AM.1n hold. Then assumptions AM.2n and AM.3n together are equivalent to

$$\lim_{s \rightarrow 0} \theta(x, p, s) = 0, \quad (11.2.1)$$

where

$$\theta(x, p, s) = \frac{f(x + s) - m(x, p, s)}{\|s\|}. \quad (11.2.2)$$

**Proof.** We first show that (11.1.1) and (11.1.2) imply (11.2.1). Suppose otherwise that there is a sequence  $\{s_k\}$  converging to zero but that

$$\theta(x, p, s_k) \geq \kappa_{\text{tnz}} \quad (11.2.3)$$

for some  $\kappa_{\text{tnz}} > 0$ . Furthermore, let

$$t_k = \|s_k\| \text{ and } d_k = \frac{s_k}{t_k}.$$

As  $\|d_k\| = 1$ , the sequence  $\{d_k\}$  is bounded and there must be a convergent subsequence  $\{d_k\}$  ( $k \in \mathcal{K}$ ) whose limit is some  $d_*$  for which  $\|d_*\| = 1$ .

It follows by definition (11.2.2) and from AM.2n that

$$\theta(x, p, s_k) = \frac{f(x + t_k d_k) - f(x)}{t_k} - \frac{m(x, p, t_k d_k) - m(x, p, 0)}{t_k}, \quad (11.2.4)$$

where  $t_k$  converges to zero. Examining the first term of (11.2.4), we have that

$$\frac{f(x + t_k d_k) - f(x)}{t_k} = \frac{f(x + t_k d_k) - f(x + t_k d_*)}{t_k} + \frac{f(x + t_k d_*) - f(x)}{t_k} \quad (11.2.5)$$

for  $k \in \mathcal{K}$ . The local Lipschitz continuity of  $f$  (AF.1n) implies that the first term in (11.2.5) converges to zero as  $k \in \mathcal{K}$  increases, while its regularity gives that the second term in (11.2.5) converges to  $f_{d_*}^o(x)$ . Identical reasoning shows that the second term in (11.2.4) converges to  $m_{d_*}^o(x, p, 0)$ , as  $k \in \mathcal{K}$  increases, using AM.1n. Thus

$$\lim_{k \in \mathcal{K} \rightarrow \infty} \theta(x, p, s_k) = f_{d_*}^o(x) - m_{d_*}^o(x, p, 0) = 0$$

from AM.3n, which contradicts (11.2.3). Thus AM.2n and AM.3n imply (11.2.1).

It is clear that (11.2.1) is equivalent to

$$\lim_{t \searrow 0} \theta(x, p, td) = 0 \quad (11.2.6)$$

for all bounded  $d \neq 0 \in \mathbb{R}^n$ , so it remains to show that (11.2.6) implies AM.2n and AM.3n. By the definition of  $\theta(x, p, s)$ ,

$$f(x + td) = m(x, p, td) + \theta(x, p, td)t\|d\|$$

for  $t \geq 0$ . AF.1n and AM.1n imply that  $f$  and  $m$  are continuous around  $t = 0$ , and by (11.2.6) it follows that  $f(x) = m(x, p, 0)$ , which is AM.2n. It also follows from AF.1n, AM.1n, and (11.2.6) that

$$\begin{aligned} f_d^o(x) &= \lim_{t \searrow 0} \frac{f(x + td) - f(x)}{t} \\ &= \lim_{t \searrow 0} \left( \frac{m(x, p, td) - m(x, p, 0)}{t} + \theta(x, p, td)\|d\| \right) = m_d^o(x, p, 0) \end{aligned}$$

for all  $d \neq 0 \in \mathbb{R}^n$ . Thus AM.2n and AM.3n hold.  $\square$

We now show that the convergence of  $\theta$  implied in (11.2.1) is uniform in  $\mathcal{P}$ .

**Lemma 11.2.2** Suppose assumptions AF.1n, AM.1n, and AM.4n hold. Then given any  $x_*$  and  $\delta > 0$ , there are values  $\epsilon > 0$  and  $\Delta > 0$  for which  $|\theta(x, p, s)| \leq \delta$  for all  $\|s\| \leq \Delta$ , all  $x \in \mathcal{O}_\epsilon(x_*)$ , and  $p \in \mathcal{P}$  whenever the limit (11.2.1) holds.

**Proof.** Suppose otherwise that for any sequences  $\{\Delta_k\}$  and  $\{\epsilon_k\}$  of positive values whose limits are zero, there are values  $x_k$ ,  $p_k$ , and  $s_k$  for which  $\|s_k\| = \Delta_k$ ,  $x_k \in \mathcal{O}_{\epsilon_k}(x_*)$ , and

$$|\theta(x_k, p_k, s_k)| > \delta. \quad (11.2.7)$$

As  $\{p_k\}$  is bounded under assumption AM.4n, so is  $\{x_k, p_k, s_k\}$ , and thus this latter sequence must have a convergent subsequence  $\{(x_k, p_k, s_k)\}$  ( $k \in \mathcal{K}$ ) whose limit is  $(x_*, p_*, 0)$  for some  $p_* \in \mathcal{P}$ . Since AF.1n and AM.1n imply that  $\theta(x, p, s)$  is a continuous function of  $(x, p)$ , (11.2.7) gives that

$$|\theta(x_*, p_*, s_k)| > \frac{1}{2}\delta$$

with  $\|s_k\|$  converging to zero, which contradicts (11.2.1) and thus establishes the lemma.  $\square$

We next show that, if an algorithm based on a step satisfying (11.1.14) appears to stagnate in the neighbourhood of a noncritical point, the trust-region radius must eventually become so small that a successful step will occur.

**Lemma 11.2.3** Suppose that  $x_*$  is not a first-order critical point of  $f$  and that assumptions AF.1n and AM.1n–AM.4n hold. Let  $\eta \in (0, 1)$ . Then there are strictly positive values  $\Delta^{\max}$  and  $\epsilon$  such that for all  $x \in \mathcal{O}_\epsilon(x_*)$ ,  $p \in \mathcal{P}$ , and  $\Delta \in (0, \Delta^{\max})$ ,

$$\rho^c(x, p, s) \stackrel{\text{def}}{=} \frac{f(x) - f(x + s)}{m(x, p, 0) - m(x, p, s)} \geq \eta$$

for any  $s$  satisfying (11.1.14).

**Proof.** The stated assumptions and Lemma 11.2.1 are sufficient to ensure that

$$f(x + s) = m(x, p, s) + \theta(x, p, s)\|s\|,$$

where  $\theta(x, p, s)$  satisfies (11.2.1) for every  $x \in \mathcal{O}_\epsilon(x_*)$ . Hence,

$$f(x) - f(x + s) = m(x, p, 0) - m(x, p, s) - \theta(x, p, s)\|s\|,$$

and thus there are positive values  $\Delta_1^{\max}$  and  $\epsilon_1$  such that

$$\rho^c(x, p, s) = 1 - \frac{\theta(x, p, s)\|s\|}{m(x, p, 0) - m(x, p, s)} \quad (11.2.8)$$

for every  $x \in \mathcal{O}_{\epsilon_1}(x_*)$  and  $\|s\| \leq \Delta_1^{\max}$ .

Since  $x_*$  is not critical, Lemma 11.1.2 shows that  $g(x)$  is bounded away from zero in some neighbourhood of  $x_*$ , and thus we have that  $\|g(x)\| \geq \kappa_{\text{fdd}}$  for some  $\kappa_{\text{fdd}} > 0$ . Combining this inequality with (11.1.14), (11.2.8), and the inequality  $\|s\| \leq \Delta$ , we have that there are  $\Delta_2^{\max} \leq \min[\delta, \Delta_1^{\max}]$  and  $\epsilon_2 \leq \epsilon_1$  for which

$$\rho^c(x, p, s) \geq 1 - \frac{\theta(x, p, s)}{\kappa_{\text{mde}}\|g(x)\|} \frac{\|s\|}{\Delta} \geq 1 - \frac{\theta(x, p, s)}{\kappa_{\text{mde}}\kappa_{\text{fdd}}}$$

for every  $x \in \mathcal{O}_{\epsilon_2}(x_*)$  and  $\Delta \leq \Delta_2^{\max}$ . The result then follows since Lemma 11.2.2 shows that  $\theta(x, p, s)$  may be made arbitrarily small for all  $x \in \mathcal{O}_\epsilon(x_*)$  and  $p \in \mathcal{P}$  by picking  $\delta$  sufficiently small.  $\square$

The next step is, in the same vein as Theorem 6.4.3 (p. 135), to show that the trust-region radius cannot become too small if the iterates approach a noncritical point of the objective function.

**Lemma 11.2.4** Suppose that  $x_*$  is a limit point of the sequence  $\{x_k\}$  generated by Algorithm 11.1.1, that  $x_*$  is not a first-order critical point, and that assumptions AF.1n, AM.1n–AM.4n, and AA.1n hold. Then there is a constant  $\Delta_{\min} > 0$ , so that for every subsequence  $\{x_k\}$  ( $k \in \mathcal{K}$ ) which converges to  $x_*$ , we have

$$\Delta_k \geq \Delta_{\min} \quad (11.2.9)$$

for all  $k \in \mathcal{K}$ .

**Proof.** Lemma 11.2.3 implies that there are scalars

$$0 < \Delta_2^{\max} \leq \Delta_1^{\max} \text{ and } 0 < \epsilon_2 \leq \epsilon_1$$

such that

$$\rho^C(x, p, s) \geq \eta_1 \text{ for all } x \in \mathcal{O}_{\epsilon_1}(x_*) \text{ and } \Delta \in (0, \Delta_1^{\max}]$$

and

$$\rho^C(x, p, s) \geq \eta_2 \text{ for all } x \in \mathcal{O}_{\epsilon_2}(x_*) \text{ and } \Delta \in (0, \Delta_2^{\max}] \quad (11.2.10)$$

for all  $p \in \mathcal{P}$  and any  $s$  satisfying (11.1.14).

Since the iterates are unchanged on unsuccessful iterations, we remove the repeated unsuccessful iterates from the sequence  $\{x_k\}$  ( $k \in \mathcal{K}$ ), so that  $\mathcal{K}$  only consists of the indices of successful iterations. Let  $t_k \geq 0$  denote the number of unsuccessful iterations immediately prior to that which yields the successful iterate  $x_k$ , and let the trust-region radii on these iterations be  $\Delta_k^{(j)}$ ,  $j = 0, \dots, t_k$ , with  $\Delta_k = \Delta_k^{(t_k)}$ . We illustrate this in Figure 11.2.1.

...	$\Delta_{k-1}^{(t_{k-1})}$	$\mid$	$\Delta_k^{(0)}$	$\Delta_k^{(1)}$	$\dots$	$\Delta_k^{(t_k-1)}$	$\Delta_k^{(t_k)}$	$\mid$	$\Delta_{k+1}^{(0)}$	$\dots$
...	$\Delta_{k-1}$				$\dots$		$\Delta_k$			$\dots$
...	$x_{k-1}$				$\dots$		$x_k$			$\dots$
...	s		u	u	$\dots$	u	s		u	$\dots$

Figure 11.2.1: The sequence of iterates along with their trust-region radii. The letters “s” and “u” indicate successful and unsuccessful iterations, respectively.

According to (11.1.16) we have that

$$\gamma_1 \Delta_k^{(j)} \leq \Delta_k^{(j+1)} \leq \gamma_2 \Delta_k^{(j)} \quad \text{for } j = 0, \dots, t_k - 1 \quad (11.2.11)$$

and

$$\rho_k = \rho_k^{(j)} < \eta_1 \quad \text{for } j = 0, \dots, t_k - 1 \quad \text{and} \quad \rho_k^{(t_k)} \geq \eta_1,$$

which in turn implies that

$$\rho_k = \rho_k^C = \rho_{k-1}^{C(j)} < \eta_1 \quad \text{for } j = 0, \dots, t_k - 1.$$

For each successful iteration, we obtain the initial (trial) radius

$$\Delta_{k+1}^{(0)} \geq \gamma_2 \Delta_k \quad (11.2.12)$$

for the next iteration.

Consider  $k \in \mathcal{K}$  sufficiently large that  $x_k \in \mathcal{O}_{\epsilon_1}(x_*)$ . If the initial radius  $\Delta_k^{(0)} > \Delta_1^{\max}$  for all such  $k$ , then  $\rho_k^{(t_k-1)} < \eta_1$  and  $\Delta_k^{(t_k-1)} > \Delta_1^{\max}$  and thus

$$\Delta_k = \Delta_k^{(t_k)} \geq \gamma_1 \Delta_k^{(t_k-1)} > \gamma_1 \Delta_1^{\max} > 0,$$

which is the desired result. So, suppose otherwise that there is a subsequence  $\mathcal{K}_1 \subset \mathcal{K}$  so that  $\Delta_k^{(0)} \leq \Delta_1^{\max}$  for all such  $k \in \mathcal{K}_1$ . Lemma 11.2.3 implies that

$$\rho_k^{(0)} \geq \rho_{k-1}^{C(0)} \geq \eta_1,$$

and thus  $t_k = 0$  and

$$\Delta_k = \Delta_k^{(0)} \quad (11.2.13)$$

for all  $k \in \mathcal{K}_1$ .

Now suppose that

$$\lim_{k \in \mathcal{K}_1} \Delta_k = 0. \quad (11.2.14)$$

It then follows that for all  $k \in \mathcal{K}_1$  sufficiently large,  $x_k \in \mathcal{O}_{\epsilon_2}(x_*)$  and  $\Delta_k \leq \Delta_2^{\max}$ . Let  $\mathcal{K}_0$  be the set of all indices in  $\mathcal{K}_1$  that are sufficiently large so that

$$x_k \in \mathcal{O}_{\frac{1}{2}\epsilon_2}(x_*) \subset \mathcal{O}_{\epsilon_2}(x_*) \quad \text{and} \quad \Delta_k \leq \Delta_2^{\max}, \quad (11.2.15)$$

and consider the history of the trust-region radii immediately preceding iteration  $k_0 \in \mathcal{K}_0$ .

If  $x_{k_0-1}$  is not in  $\mathcal{O}_{\epsilon_2}(x_*)$ , then by definition,  $\|x_{k_0-1} - x_*\| \geq \epsilon_2$ , while from (11.2.15)  $\|x_{k_0} - x_*\| < \frac{1}{2}\epsilon_2$ . Hence, it follows from (11.2.12) and (11.2.13) that

$$\begin{aligned} \Delta_{k_0} &= \Delta_{k_0}^{(0)} \geq \gamma_2 \Delta_{k_0-1} \geq \gamma_2 \|s_{k_0-1}\| = \gamma_2 \|x_{k_0-1} - x_{k_0}\| \\ &\geq \gamma_2 \|x_{k_0-1} - x_*\| - \gamma_2 \|x_{k_0} - x_*\| > \frac{1}{2}\gamma_2 \epsilon_2. \end{aligned} \quad (11.2.16)$$

As this contradicts (11.2.14), it follows that  $x_{k_0-1}$  must also be in  $\mathcal{O}_{\epsilon_2}(x_*)$ .

Suppose now that  $\{x_{k_0-\ell}, \dots, x_{k_0}\} \in \mathcal{O}_{\epsilon_2}(x_*)$  for some  $\ell \geq 1$ , but that  $x_{k_0-\ell-1}$  is not in  $\mathcal{O}_{\epsilon_2}(x_*)$ . It follows from (11.2.12) that

$$\Delta_{k_0-i+1}^{(0)} \geq \gamma_2 \Delta_{k_0-i} \text{ for } i = 1, \dots, \ell.$$

If one or more of the trust-region radii is not increased, there must be a smallest index  $i \in \{1, \dots, \ell\}$  for which

$$\Delta_{k_0-i} \leq \Delta_{k_0-i}^{(0)}$$

and

$$\Delta_{k_0} > \Delta_{k_0-1} > \dots > \Delta_{k_0-i}. \quad (11.2.17)$$

Suppose further that  $i < \ell$ . In this case,  $x_{k_0-i} \in \mathcal{O}_{\epsilon_2}(x_*)$  and it must be that

$$\Delta_{k_0-i} > \gamma_2 \Delta_2^{\max}. \quad (11.2.18)$$

To see this, suppose otherwise that  $\Delta_{k_0-i-1} < \Delta_2^{\max}$ . Then because  $x_{k_0-i} \in \mathcal{O}_{\epsilon_2}(x_*)$ , Lemma 11.2.3 gives that  $\rho_{k_0-i-1}^C \geq \eta_2$  and thus  $\rho_{k_0-i-1} \geq \eta_2$ . Hence iteration  $k_0 - i - 1$  is very successful and we obtain from (11.1.16) that  $\Delta_{k_0-i} \geq \gamma_3 \Delta_{k_0-i-1} > \Delta_{k_0-i-1}$ , which contradicts the fact that  $i$  is smallest. Thus (11.2.18) must hold. But in this case, (11.2.17) and (11.2.18) give that

$$\Delta_{k_0} > \gamma_2 \Delta_2^{\max},$$

which contradicts (11.2.14). Consequently  $i = \ell$ , and it follows from (11.2.15) and (11.2.17) that

$$\Delta_2^{\max} \geq \Delta_{k_0} > \Delta_{k_0-1} > \dots > \Delta_{k_0-\ell}. \quad (11.2.19)$$

Hence  $x_{k_0-i} \in \mathcal{O}_{\epsilon_2}(x_*)$  and  $\Delta_{k_0-i} < \Delta_2^{\max}$  for all  $0 \leq i \leq \ell$ , and it then follows from (11.2.10) that

$$\rho_{k_0-i} \geq \rho_{k_0-i}^C > \eta_2,$$

and so each of these iterations is very successful. Thus

$$\Delta_{k_0-i+1} \geq \gamma_3 \Delta_{k_0-i} \text{ for } i = 1, \dots, \ell;$$

that is,

$$\Delta_{k_0-i} \leq \frac{\Delta_{k_0}}{\gamma_3^i} \text{ for } i = 1, \dots, \ell. \quad (11.2.20)$$

We now consider the progress of the trust-region radius between iterations  $k_0 - \ell - 1$  and  $k_0 - \ell$ . We have from (11.2.12) that

$$\Delta_{k_0-\ell}^{(0)} \geq \gamma_2 \Delta_{k_0-\ell-1}. \quad (11.2.21)$$

Suppose that  $\Delta_{k_0-\ell}^{(0)} > \Delta_{k_0}$ . In this case, there must be a smallest integer  $s$  for which

$$\Delta_{k_0-\ell}^{(s-1)} > \Delta_{k_0} \text{ and } \Delta_{k_0-\ell}^{(s-1)} \leq \Delta_{k_0},$$

since  $\Delta_{k_0-\ell}^{(t_{k_0-\ell})} = \Delta_{k_0-\ell} < \Delta_2^{\max}$ . Now, (11.2.11) and (11.2.11) imply that

$$\Delta_{k_0-\ell}^{(s)} \geq \gamma_1 \Delta_{k_0-\ell}^{(s-1)} > \gamma_1 \Delta_{k_0}. \quad (11.2.22)$$

But as  $\Delta_{k_0-\ell}^{(s-1)} \leq \Delta_{k_0} \leq \Delta_2^{\max}$  and  $x_{k_0-\ell} \in \mathcal{O}_{\epsilon_2}(x_*)$ , Lemma 11.2.3 implies that  $s = t_k$  and thus (11.2.20) gives that

$$\Delta_{k_0-\ell}^{(s)} = \Delta_{k_0-\ell} \leq \frac{\Delta_{k_0}}{\gamma_3^\ell}. \quad (11.2.23)$$

Combining (11.2.22) and (11.2.23), we deduce that  $\gamma_1 < 1/\gamma_3^\ell$ , which contradicts (11.1.15), and thus we must have that  $\Delta_{k_0-\ell}^{(0)} \leq \Delta_{k_0}$ . Consequently,

$$\Delta_{k_0-\ell}^{(0)} \leq \Delta_{k_0} \leq \Delta_2^{\max}$$

and  $x_{k_0-\ell} \in \mathcal{O}_{\epsilon_2}(x_*)$ . Lemma 11.2.3 implies that  $\Delta_{k_0-\ell}^{(0)} = \Delta_{k_0-\ell}$ , and hence, from (11.2.19) and (11.2.21), that

$$\Delta_{k_0-\ell-1} \leq \frac{\Delta_{k_0-\ell}}{\gamma_2} \leq \frac{\Delta_{k_0}}{\gamma_2}. \quad (11.2.24)$$

Therefore, combining (11.2.20) and (11.2.24), we deduce that

$$\sum_{i=0}^{\ell+1} \Delta_{k_0-i} \leq \frac{\Delta_{k_0}}{\gamma_2} + \sum_{i=0}^{\ell} \frac{\Delta_{k_0}}{\gamma_3^i} \leq \frac{\Delta_{k_0}}{\gamma_2} + \Delta_{k_0} \sum_{i=0}^{\infty} \left( \frac{1}{\gamma_3} \right)^i = \left( \frac{1}{\gamma_2} + \frac{1}{\gamma_3 - 1} \right) \Delta_{k_0}. \quad (11.2.25)$$

But, as  $x_{k_0-\ell-1}$  is not in  $\mathcal{O}_{\epsilon_2}(x_*)$ , by definition  $\|x_{k_0-\ell-1} - x_*\| \geq \epsilon_2$ , while from (11.2.15)  $\|x_{k_0} - x_*\| < \frac{1}{2}\epsilon_2$ . Hence,

$$\begin{aligned} \sum_{i=0}^{\ell+1} \Delta_{k_0-i} &\geq \sum_{i=0}^{\ell+1} \|s_{k_0-i}\| \geq \left\| \sum_{i=0}^{\ell+1} s_{k_0-i} \right\| = \|x_{k_0} - x_{k_0-\ell-1}\| \\ &\geq \|x_{k_0-\ell-1} - x_*\| - \|x_{k_0} - x_*\| > \frac{1}{2}\epsilon_2. \end{aligned} \quad (11.2.26)$$

Combining (11.2.25) and (11.2.26) implies that

$$\Delta_{k_0} \geq \epsilon_2/2 \left( \frac{1}{\gamma_2} + \frac{1}{\gamma_3 - 1} \right) > 0,$$

which contradicts (11.2.14). Therefore  $x_i \in \mathcal{O}_{\epsilon_2}(x_*)$  for all  $i \leq k_0$ , and hence  $\Delta_i \geq \gamma_1 \Delta_2^{\max}$ . That this is true for all  $k_0$  in  $\mathcal{K}_0$  contradicts (11.2.14), and thus (11.2.9) must hold for some  $\Delta_{\min} > 0$ .  $\square$

Finally, we are in a position to prove our general global convergence result for nonsmooth optimization.

**Theorem 11.2.5** Suppose that  $x_*$  is a limit point of the sequence  $\{x_k\}$  generated by Algorithm 11.1.1 and that assumptions AF.1n, AM.1n–AM.4n, and AA.1n hold. Then  $x_*$  is a first-order critical point of  $f(x)$ .

**Proof.** Suppose otherwise that  $x_*$  is not first-order critical. Let  $\mathcal{K}$  be the indices of any subsequence of successful iterations for which  $\{x_k\}$  converges to  $x_*$ . Since  $x_*$  is not first-order critical, Lemma 11.2.4 implies that there is a  $\Delta_{\min} > 0$  such that (11.2.9) holds for all  $k \in \mathcal{K}$ . As iteration  $k \in \mathcal{K}$  is successful, we have that  $\rho_k \geq \eta_1$ , which is to say that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 [m(x_k, p_k, 0) - m(x_k, p_k, s_k)]. \quad (11.2.27)$$

But, under assumption AA.1n,

$$\begin{aligned} m(x_k, p_k, 0) - m(x_k, p_k, s_k) &\geq \kappa_{\text{mdc}} \|g(x_k)\| \min [\delta_k, \Delta_k] \\ &\geq \kappa_{\text{mdc}} \|g(x_k)\| \min [\delta, \Delta_{\min}]. \end{aligned}$$

Combining this with (11.2.27), we find that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 \kappa_{\text{mdc}} \min [\delta, \Delta_{\min}] \|g(x_k)\|$$

for all  $k \in \mathcal{K}$ . But then, if we let  $\mathcal{K} = \{k_i\}$  for  $i \geq 0$ ,

$$\sum_{i=0}^{\infty} [f(x_{k_i}) - f(x_{k_i+1})] \leq \sum_{i=0}^{\infty} [f(x_{k_i}) - f(x_{k_i+1})] = f(x_{k_0}) - f(x_*),$$

and thus  $f(x_{k_i}) - f(x_{k_i+1})$  and hence  $\|g(x_k)\|$  must converge to zero. But this contradicts the assumption that  $x_*$  is not first-order critical, as Lemma 11.1.2 then implies that  $\|g(x)\|$  is bounded away from zero in some neighbourhood of  $x_*$ . Thus  $x_*$  must be first-order critical.  $\square$

It is worth comparing this result with the convergence results for smooth problems given in Section 6.4. The result here makes an assumption that the sequence  $\{x_k\}$  has a limit point, while no such assumption is made in Section 6.4. On the other hand, the results here make no use of Taylor's theorem, and in particular, the error between model and function is not required to depend quadratically on the trust-region radius as is implied by Theorem 6.4.1 (p. 133). The boundedness of the parameters here (AM.4n) is consistent with the requirement AM.4 in Section 6.4, but of course we have noted in Section 8.4 that AM.4 may be relaxed to AM.4d, and we have remarked in this section that AM.4n may be too strong even for nonsmooth problems.

We should stress that, while the results given here are stated for simplicity in terms of an  $\ell_2$ -norm trust region, the reader may readily verify that they are equally applicable for any (perhaps iteration-dependent) set of uniformly equivalent norms.

## Notes and References for Section 11.2

The material in this section is heavily based upon Dennis, Li, and Tapia (1995), whose algorithm is similar to that analysed here except that it relies upon a much stronger step requirement, namely, that

$$m(x_k, p_k, 0) - m(x_k, p_k, s_k) \geq \kappa_{\text{mdc}} [m(x_k, p_k, 0) - m(x_k, p_k, s_k^M)] \text{ and } \|s_k\| \leq \Delta$$

for some  $\kappa_{\text{mdc}} \in (0, 1]$  and all  $k$ , where  $s^M$  is a global minimizer of (11.1.7). For the slightly weaker class of polyhedral convex functions, this method was predated by those proposed by Fletcher (1982a, 1987a, Chapter 14) and Yamakawa, Fukushima, and Ibaraki (1989), in which an exact minimization of the model problem is performed, and those of Wright (1989b, 1990), in which an inexact solution, satisfying conditions similar to those we have considered, is used. Variations and subtle improvements on Dennis, Li, and Tapia's (1995) analysis, in which different nonsmoothness assumptions are made on the objective function and its model, are given by Qi and Sun (1994), Poliquin and Qi (1995), and de Sampaio, Yuan, and Sun (1997). A more significant development, by Scholtes and Stöhr (1999), replaces the assumption that  $f$  be regular with the more general assumption that the one-sided directional derivative of  $f$  always exists. They show that, if the trust-region radius is bounded away from zero, any limit point  $x_*$  of their variant of Algorithm 11.1.1 is such that  $f'_d(x_*) \geq 0$ , while otherwise at least  $0 \in \partial f(x_*)$ . Of course for regular functions the two conclusions are the same. For irregular functions, the former result is significantly more powerful. Stöhr (1999), Scholtes and Stöhr (1999), and Colson (1999) provide a discussion of the application of the resulting method to mathematical programming problems with equilibrium constraints (MPECs).

For highly irregular functions, the above analysis is too specific. For such problems, the most important class of methods is known as bundle methods, of which there are trust-region variants called *trust-bundle* methods. The idea is best explained for nonsmooth convex functions and operates under the reasonable assumption that, although the entire generalized gradient (or, in the convex case, subdifferential) might be unknown, a single (perhaps random) element (or subgradient) can often be found. Since the objective function  $f(x)$  is convex, we have that

$$f(x_i) + \langle g_i, x - x_i \rangle \leq f(x),$$

where  $g_i$  is a subgradient at  $x_i$ , for all  $x$ . Suppose that we have computed the function values  $f(x_i)$  and representative subgradients  $g_i$  for a sequence of points  $x_i$ ,  $0 \leq i \leq k$ . Then a trust-bundle method has an number of ingredients. Firstly, the model

$$m_k(x) = \max_{0 \leq i \leq k} f(x_i) + \langle g_i, x - x_i \rangle$$

provides an increasing lower bound on  $f(x)$  and agrees with  $f(x)$  at  $x_i$ ,  $0 \leq i \leq k$ . Secondly, since this model may be unbounded from below, it is only considered to be useful within an appropriate trust region  $\|x - x_{s(k)}\| \leq \Delta_k$  for some appropriate norm  $\|\cdot\|$  and radius  $\Delta_k$ . Here,  $s(k)$  is the last index on which the objective function was reduced by a significant amount. In the terminology with which we are familiar,  $s(k)$  is the last successful iteration, although in trust-bundle terminology this is usually known as a *descent* or *serious* step—unsuccessful steps are known as *null* steps. Finally, the next iterate  $x_{k+1}$  is obtained by (approximately) minimizing  $m_k(x)$  within the trust region  $\|x - x_{s(k)}\| \leq \Delta_k$ , and the iteration is deemed a success if and only if

$$\frac{f(x_{s(k)}) - f(x_{k+1})}{m_k(x_{s(k)}) - m_k(x_{k+1})} \geq \eta_1$$

for some  $\eta_1$ . The iteration is said to have converged when  $f(x_{k+1}) - m_k(x_{k+1})$  is small. Interestingly, the convergence of the method is ensured not by reducing the radius following an unsuccessful step, but by improving the model by means of an additional segment  $f(x_{k+1}) + \langle g_{k+1}, x - x_{k+1} \rangle$ . This has some similarities with the conditional models we considered in Chapter 9. See Kiwiel (1989a, 1989b), Hiriart-Urruty and Lemaréchal (1993a, 1993b), Schramm and Zowe (1992), Lemaréchal and Zowe (1994), Fukushima et al. (1996), Grothey

and McKinnon (1998), Einarsson and Madsen (1998), Fei and Huachen (1998), and Frangioni and Gallo (1999) for details of both the convex and nonconvex cases. The software package BT (Outrata, Zowe, and Schramm, 1991) uses just such a method.

## 11.3 Variations on This Theme

We next consider a couple of important variations on the themes of Sections 11.1 and 11.2.

### 11.3.1 A Nonmonotonic Variant

In Section 10.1, we considered an important variant of our basic trust-region method in which the requirement of a monotonic decrease of the objective function is replaced by a requirement that there is a monotonically downward trend. In this section we show that such a variant is also possible when the objective function is nonsmooth. We shall see in Section 15.3.2 that this additional flexibility proves to be extremely important when minimizing nonsmooth exact penalty functions for constrained optimization.

The extension to the nonmonotonic variant is achieved very simply by replacing the requirement that the step calculated in Step 2 of Algorithm 10.1.1 (p. 349) satisfies AA.1 by the nonsmooth step requirement AA.1n. As before, we also need to strengthen the trust-region radius update conditions in Step 4 from (6.1.5) (p. 116) to (11.1.16). It is easy to see that most of the theory we developed in Section 11.1 goes through unchanged under this modification. The proof of Lemma 11.2.4 depends on the choice of  $\rho_k$ , but the fact that  $\rho_k \geq \rho_k^C = \rho(x_k, p_k, s_k)$  and the consequences of Lemma 11.2.3 reveals that the same method of proof applies. It thus remains for us to consider the appropriate global convergence result for nonsmooth nonmonotone trust-region methods.

**Theorem 11.3.1** Suppose that  $x_*$  is a limit point of the sequence  $\{x_k\}$  generated by Algorithm 10.1.1 (with (11.1.16) replacing (6.1.5)), and that assumptions AF.1n, AF.2, AM.1n–AM.4n, and AA.1n hold. Then  $x_*$  is a first-order critical point of  $f(x)$ .

**Proof.** Let  $\mathcal{K}$  be the indices of any subsequence of successful iterations for which  $\{x_k\}$  converges to  $x_*$ . Suppose there are only a finite number of successful iterations indexed by  $\mathcal{K}$ , but that  $x_*$  is not critical. Let  $k_0$  be the final successful iteration. Then  $x_k = x_{k_0}$  for all  $k > k_0$ , and hence the algorithm requires that the trust-region radius must converge to zero. But it then follows from Lemma 11.2.3 that  $\rho_{k_1}^C \geq \eta_1$  for sufficiently large  $k_1 > k_0$ , and thus iteration  $k_1$  is successful. Since this contradicts the maximality of  $k_0$ ,  $x_*$  must be critical.

So now suppose that the number of successful iterations in  $\mathcal{K}$  is infinite, but that  $x_*$  is not first-order critical. Lemma 11.1.2 then implies that there exists an  $\epsilon > 0$  such that

$$\|g(x)\| \geq \epsilon$$

in some neighbourhood of  $x_*$ . Given the successful iterate  $x_{k+1}$ , we use the backward referencing sequence of successful iterates (10.1.6) (p. 350), which we introduced in the proof of Lemma 10.1.1 (p. 350), and immediately obtain that

$$\begin{aligned} f(x_0) - f(x_{k+1}) &= \sum_{j=1}^q [f(x_{p_{j-1}+1}) - f(x_{p_j+1})] + f(x_{p_k}) - f(x_{k+1}) \\ &\geq \eta_1 \sum_{\substack{t=0 \\ t \in \mathcal{S}}}^k [m(x_t, p_t, 0) - m(x_t, p_t, s_t)] \\ &\geq \eta_1 \sum_{\substack{t=k_0 \\ t \in \mathcal{S} \cap \mathcal{K}}}^k [m(x_t, p_t, 0) - m(x_t, p_t, s_t)] \end{aligned} \quad (11.3.1)$$

for any  $k_0 < k$ , using the acceptance rule from Step 3 of the algorithm, and the fact that each term  $m(x_t, p_t, 0) - m(x_t, p_t, s_t)$  is positive. Since  $x_*$  is not first-order critical, the appropriate version of Lemma 11.2.4 for the nonmonotone case implies that there is a  $\Delta_{\min} > 0$  such that (11.2.9) holds for all  $k \in \mathcal{K}$ . But, using AA.1n and because  $x_*$  is not critical,

$$m(x_t, p_t, 0) - m(x_t, p_t, s_t) \geq \kappa_{\text{mdc}} \epsilon \min [\delta, \Delta_{\min}] \quad (11.3.2)$$

for all  $t \in \mathcal{S} \cap \mathcal{K}$  sufficiently large, say, greater than  $k_0$ . Thus, combining (11.3.1) and (11.3.2), it follows immediately that

$$f(x_0) - f(x_{k+1}) \geq \nu_k \eta_1 \kappa_{\text{mdc}} \epsilon \min [\delta, \Delta_{\min}] > 0, \quad (11.3.3)$$

where

$$\nu_k \stackrel{\text{def}}{=} |\mathcal{S} \cap \mathcal{K} \cap \{k_0, \dots, k\}|$$

is the number of successful iterations in  $\mathcal{K}$  following  $k_0$  and up to and including iteration  $k$ . Since the objective function is bounded below by AF.2, the left-hand side of (11.3.3) is bounded by a quantity independent of  $k$ , and  $\nu_k$  must therefore satisfy

$$\nu_k \leq \nu_{\max}$$

for some  $\nu_{\max}$  independent of  $k$ . But this contradicts the assumption that there are infinitely many successful iterations in  $\mathcal{K}$ . Thus  $x_*$  must be first-order critical.  $\square$

### 11.3.2 Correction Steps

We gave a second important variant of the basic trust-region method in Section 10.4.2 by allowing the use of magical steps to further reduce the objective function. Of particular interest to us here is the special case of correction steps discussed in Section 10.4.2. It is very easy to state just such an algorithm for nonsmooth problems.

**Algorithm 11.3.1: Nonsmooth algorithm with corrections**

**Step 0: Initialization.** An initial point  $x_0$ , initial and maximum permissible trust-region radii  $0 < \Delta_0 \leq \Delta_{\max}$ , and an enlargement factor  $\tau \geq 1$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  are also given and satisfy the conditions

$$0 < \eta_1 \leq \eta_2 < 1 \text{ and } 0 < 1/\gamma_3 \leq \gamma_1 \leq \gamma_2 < 1 < \gamma_3.$$

Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k$  and define a model  $m(x_k, p_k, s)$  in  $\mathcal{B}_k$  satisfying AM.1n–AM.4n.

**Step 2: Step calculation.** Compute a step  $s_k$  satisfying AA.1n and for which  $x_k + s_k \in \mathcal{B}_k$ . Determine a correction step  $s_k^{\text{CS}}$  for which both

$$\|s_k + s_k^{\text{CS}}\|_k \leq \tau \Delta_k \quad (11.3.4)$$

and

$$f(x_k + s_k + s_k^{\text{CS}}) \leq f(x_k + s_k). \quad (11.3.5)$$

**Step 3: Acceptance of the trial point.** Compute the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k + s_k^{\text{CS}})}{m(x_k, p_k, 0) - m(x_k, p_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k + s_k^{\text{CS}}$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\gamma_3 \Delta_k, \Delta_{\max}] & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1, \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

Notice that the choice  $s_k^{\text{CS}} = 0$  is allowed, in which case Algorithm 11.3.1 is equivalent to its predecessor, Algorithm 11.1.1. Of more significance is that there is now no requirement that the step  $s_k$  actually decrease the objective function even on successful iterations, merely that  $s_k + s_k^{\text{CS}}$  do so. Moreover, we do not require that the correction step  $s_k^{\text{CS}}$  remain within the original trust region so long as (11.3.4) holds, and this gives us further flexibility. All of these observations will turn out to be extremely important in Section 15.3.2 when minimizing nonsmooth exact penalty functions for constrained optimization.

The analysis of the new algorithm is straightforward. A careful examination of the proofs of the results in Section 11.2 shows that Lemma 11.2.3 remains true if  $\rho^C(x, p, s)$

is replaced by

$$\rho^C(x, p, s, s^{CS}) = \frac{f(x) - f(x + s + s^{CS})}{m(x, p, 0) - m(x, p, s)},$$

so long as  $f(x + s + s^{CS}) \leq f(x + s)$ , since then

$$\rho^C(x, p, s, s^{CS}) \geq \frac{f(x) - f(x + s)}{m(x, p, 0) - m(x, p, s)} \geq \eta,$$

the last inequality following from Lemma 11.2.3. This will be the case in Algorithm 11.3.1 because of the requirement (11.3.5). The proof of Lemma 11.2.4 is also valid so long as we replace  $\rho^C(x, p, s)$  by  $\rho^C(x, p, s, s^{CS})$ , and  $s_k$  in (11.2.16) and (11.2.26) by  $s_k + s_k^{CS}$ ; the  $\Delta_{k_0-i}$  terms in these latter inequalities are merely replaced by  $\tau\Delta_{k_0-i}$ , because of the requirement (11.3.4), which does not effect the conclusion. Finally, the proof of Theorem 11.2.5 is unaltered. In summary we have the following theorem.

**Theorem 11.3.2** Suppose that  $x_*$  is a limit point of the sequence  $\{x_k\}$  generated by Algorithm 11.3.1, and that assumptions AF.1n, AM.1n–AM.4n, and AA.1n hold. Then  $x_*$  is a first-order critical point of  $f(x)$ .

## Notes and References for Section 11.3

The idea of using a (second-order) correction is due to Fletcher (1982b) (see also Yamakawa, Fukushima, and Ibaraki, 1989). Such a correction is shown to be necessary by Yuan (1984), since otherwise the trust-region radius may always be active, and fast convergence impossible. Yuan (1985b) shows that a variant of Fletcher's (1982b) method may converge at an asymptotic Q-superlinear rate under suitable second-order sufficiency conditions. Bannert (1994) provides an alternative in which the second-order correction is not needed, but another acceptance function is used to see whether the step direction is suitable despite not giving descent for the merit function.

## 11.4 Computing the Generalized Gradient

While the algorithms analysed in Sections 11.1–11.3 are extremely general, they do require knowledge of the structure of the generalized gradient of  $f$ . Although we know that this set is convex, it is, in general, difficult to say more without considering extremely useful, but nevertheless special, cases. In this section, we shall consider two such cases.

In both cases, the objective function is a composite function (which we discussed in Section 3.1.4) of the form

$$f(x) + h(c(x)),$$

where  $f$  and  $c$  are differentiable and  $h$  is locally Lipschitz. Since (3.1.6) (p. 34) shows that the generalized gradient of  $f(x) + h(c(x))$  depends crucially on that of  $h$ , it is the latter that is our focus here.

The first case we consider is where we partition  $c$  as

$$c = \begin{pmatrix} c_{\mathcal{E}} \\ c_{\mathcal{I}} \end{pmatrix} \quad (11.4.1)$$

and  $h(c)$  may be expressed as either

$$h(c) = h_+(c) \stackrel{\text{def}}{=} \left\| \begin{pmatrix} c_{\mathcal{E}} \\ c_{\mathcal{I}}^+ \end{pmatrix} \right\| \quad (11.4.2)$$

or

$$h(c) = h_-(c) \stackrel{\text{def}}{=} \left\| \begin{pmatrix} c_{\mathcal{E}} \\ c_{\mathcal{I}}^- \end{pmatrix} \right\|, \quad (11.4.3)$$

where the superscripts  $+$  and  $-$  denote, respectively, the componentwise maximum and minimum of a vector  $d$  and zero, viz.,

$$d^+ = \max[d, 0] \text{ and } d^- = \min[d, 0].$$

For brevity, we shall write

$$c^{\mathcal{I}+} \stackrel{\text{def}}{=} \begin{pmatrix} c_{\mathcal{E}} \\ c_{\mathcal{I}}^+ \end{pmatrix} \text{ and } c^{\mathcal{I}-} \stackrel{\text{def}}{=} \begin{pmatrix} c_{\mathcal{E}} \\ c_{\mathcal{I}}^- \end{pmatrix}.$$

We are then able to write (11.4.2) as  $h_+(c) = \|c^{\mathcal{I}+}\|$  and (11.4.3) as  $h_-(c) = \|c^{\mathcal{I}-}\|$ . Particular special cases of (11.4.2) arise when  $\mathcal{I} = \emptyset$ , in which case  $h(c) = \|c\|$ , or when  $\mathcal{E} = \emptyset$ , for which  $h(c) = \|c^+\|$ .

The second important class of problems includes those where  $h$  is a polyhedral convex function of the form

$$h(c) = h_p(c) \stackrel{\text{def}}{=} \max_{1 \leq i \leq \ell} \langle p_i, c \rangle + b_i \quad (11.4.4)$$

for given vectors  $\{p_i\}_{i=1}^\ell$  and scalars  $\{b_i\}_{i=1}^\ell$ . Some functions, such as those involving the  $\ell_1$  and  $\ell_\infty$  norms,<sup>184</sup> belong in either case. Others, such as those involving the  $\ell_2$  norm, are just covered by the first case, while functions like

$$h(c) = \max_{1 \leq i \leq m} c_i$$

belong to the second. We shall now consider how to compute the generalized gradient of functions with these forms.

We start with functions of the form (11.4.2).

---

<sup>184</sup>For  $h(c) = \|c\|_1$ ,  $\ell = 2m$  and  $p_i, p_{m+i} = \pm e_i$  and  $b_i = 0$  for  $1 \leq i \leq m$ . For  $h(c) = \|c\|_\infty$ ,  $\ell = 2^m$ , and  $p_i = \sum_{j=1}^m \sigma_j e_j$  and  $b_i = 0$ , with each  $\sigma_j = \pm 1$ .

**Theorem 11.4.1** If  $\|\cdot\|$  is monotonic and  $c$  may be partitioned as (11.4.1), the generalized gradient of the function (11.4.2) is given by

$$\partial h_+(c) = \left\{ y = \begin{pmatrix} y_{\mathcal{E}} \\ y_{\mathcal{I}} \end{pmatrix} \mid \langle y, c \rangle = h_+(c), \quad y_{\mathcal{I}} \geq 0 \text{ and } \|y\|_{\mathbb{D}} \leq 1 \right\}, \quad (11.4.5)$$

where the subscript D indicates the vector dual norm (2.3.1) (p. 21). Furthermore, if  $y \in \partial h_+(c)$ , then if  $j \in \mathcal{I}$  and  $c_j < 0$ , it follows that  $y_j = 0$ , and if  $h_+(c) > 0$ , then  $\|y\|_{\mathbb{D}} = 1$ .

**Proof.** By definition, if  $\theta \in [0, 1]$ ,

$$|c_{\mathcal{E}} + (1 - \theta)d_{\mathcal{E}}| \leq |c_{\mathcal{E}}| + (1 - \theta)|d_{\mathcal{E}}| \text{ and } (c_{\mathcal{I}} + (1 - \theta)d_{\mathcal{I}})^+ \leq c_{\mathcal{I}}^+ + (1 - \theta)d_{\mathcal{I}}^+,$$

and hence

$$\|(c + (1 - \theta)d)^{\mathcal{I}^+}\| \leq \|c^{\mathcal{I}^+} + (1 - \theta)d^{\mathcal{I}^+}\| \leq \|c^{\mathcal{I}^+}\| + (1 - \theta)\|d^{\mathcal{I}^+}\| \quad (11.4.6)$$

because  $\|\cdot\|$  is monotonic. Thus  $h_+(c) = \|c^{\mathcal{I}^+}\|$  is a convex function of  $c$ , and we then have from (3.1.3) (p. 33) that

$$\partial h_+(c) = \{y \in \mathbb{R}^n \mid \|c^{\mathcal{I}^+}\| + \langle y, d \rangle \leq \|(c + d)^{\mathcal{I}^+}\| \text{ for all } d \in \mathbb{R}^n\}. \quad (11.4.7)$$

Suppose  $y$  lies in the set (11.4.7). Then the triangle inequality and (11.4.6) (with  $\theta = 0$ ) show that

$$\|c^{\mathcal{I}^+}\| + \langle y, d \rangle \leq \|(c + d)^{\mathcal{I}^+}\| \leq \|c^{\mathcal{I}^+}\| + \|d^{\mathcal{I}^+}\|$$

and hence, since  $|d_{\mathcal{I}}^+| \leq |d_{\mathcal{I}}|$  and the norm is monotonic, that

$$\langle y, d \rangle \leq \|d^{\mathcal{I}^+}\| \leq \|d\| \quad (11.4.8)$$

for all  $d \in \mathbb{R}^n$ . Thus, by definition (2.3.1) (p. 21) and by (11.4.8),

$$\|y\|_{\mathbb{D}} = \sup_{d \neq 0} \frac{|\langle y, d \rangle|}{\|d\|} \leq 1.$$

Moreover, picking  $d = c$  in (11.4.8) together with the definition (11.4.2) gives that  $\langle y, c \rangle \leq h_+(c)$ , while setting  $d = -c$  in (11.4.7) along with (11.4.2) reveals that  $h_+(c) \leq \langle y, c \rangle$ , and thus that

$$\langle y, c \rangle = h_+(c).$$

Finally, picking  $d_{\mathcal{E}} = 0$  and  $d_{\mathcal{I}} = -e_i$  in (11.4.8) gives that  $y_i \geq 0$  for all  $i \in \mathcal{I}$ . Therefore,  $y$  lies in the set (11.4.5).

Conversely, suppose that  $y$  lies in the set (11.4.5). Then the Hölder inequality gives that

$$\begin{aligned} h_+(c) + \langle y, d \rangle &= \langle y, c \rangle + \langle y, d \rangle \\ &= \langle y, c + d \rangle \\ &\leq \langle y, (c + d)^{\mathcal{I}^+} \rangle \\ &\leq \| (c + d)^{\mathcal{I}^+} \| \| y \|_{\mathbb{D}} \\ &\leq \| (c + d)^{\mathcal{I}^+} \| \end{aligned}$$

and thus  $y$  lies in the set (11.4.7). Consequently (11.4.5) and (11.4.7) are identical.

Finally, suppose that we partition

$$c = \begin{pmatrix} c_a \\ c_b \end{pmatrix} \text{ and } y = \begin{pmatrix} y_a \\ y_b \end{pmatrix},$$

and that we let

$$c_0 = \begin{pmatrix} c_a \\ 0 \end{pmatrix} \text{ and } y_0 = \begin{pmatrix} y_a \\ 0 \end{pmatrix}.$$

Now suppose we define the norm  $\| \cdot \|_0$  as  $\| c_a \|_0 \stackrel{\text{def}}{=} \| c_0 \|$ . Then

$$\begin{aligned} \| y_a \|_{0\mathbb{D}} &= \max_{c_a} \frac{|\langle y_a, c_a \rangle|}{\| c_a \|_0} = \max_{c_a} \frac{|\langle y_0, c_0 \rangle|}{\| c_0 \|} = \max_{c_a} \frac{|\langle y, c_0 \rangle|}{\| c_0 \|} \\ &\leq \max_c \frac{|\langle y, c \rangle|}{\| c \|} = \| y \|_{\mathbb{D}}. \end{aligned} \quad (11.4.9)$$

Now consider the generalized gradient  $\partial h_+(c)$  of the function (11.4.2). Define the vector  $c_b$  to be made up of those components of  $c$  for which  $j \in \mathcal{I}$  and  $c_j < 0$ , and  $c_a$  to be the remaining components of  $c$ . Then, if  $y \in \partial h_+(c)$ , (11.4.5), (11.4.9), and the Hölder inequality show that

$$\begin{aligned} \| c_a \|_0 &= h_+(c) = \langle y, c \rangle = \langle y_a, c_a \rangle + \langle y_b, c_b \rangle \\ &\leq \| c_a \|_0 \| y_a \|_{0\mathbb{D}} + \langle y_b, c_b \rangle \leq \| c_a \|_0 \| y \|_{\mathbb{D}} + \langle y_b, c_b \rangle \\ &\leq \| c_a \|_0 + \langle y_b, c_b \rangle \end{aligned} \quad (11.4.10)$$

and thus that  $\langle y_b, c_b \rangle \geq 0$ . But since  $c_b < 0$  and  $y_b \geq 0$ , it must be that  $y_b = 0$ , which is the desired result. If  $h_+(c) > 0$ , (11.4.10) and  $y_b = 0$  give that

$$0 < h_+(c) = \| c_a \|_0 \leq \| c_a \|_0 \| y \|_{\mathbb{D}} = h_+(c) \| y \|_{\mathbb{D}}$$

and thus  $\| y \|_{\mathbb{D}} \geq 1$ . The final result then follows since  $\| y \|_{\mathbb{D}} \leq 1$ .  $\square$

It is worth considering a few special cases. Each is a straightforward application of Theorem 11.4.1, bearing in mind that the  $\ell_1$  and  $\ell_\infty$  norms are duals of each other, while the  $\ell_2$  norm is self dual (see Section 2.3.1).

**Example 11.4.1** Suppose  $h_+(c)$  is given by (11.4.2) with  $\|\cdot\| = \|\cdot\|_1$ . Then

$$\partial h_+(c) = \left\{ y \mid y_i = \begin{cases} 1 & \text{if } c_i > 0, \\ -1 & \text{if } c_i < 0 \text{ and } i \in \mathcal{E}, \\ 0 & \text{if } c_i < 0 \text{ and } i \in \mathcal{I}, \\ \in [-1, 1] & \text{if } c_i = 0 \text{ and } i \in \mathcal{E}, \\ \in [0, 1] & \text{if } c_i = 0 \text{ and } i \in \mathcal{I}. \end{cases} \right\}$$

**Example 11.4.2** Suppose  $h_+(c)$  is given by (11.4.2) with  $\|\cdot\| = \|\cdot\|_2$ . Then

$$\partial h_+(c) = \{y \mid \|y\|_2 \leq 1, \quad y_{\mathcal{I}} \geq 0 \text{ and } y = c^{\mathcal{I}^+} / \|c^{\mathcal{I}^+}\|_2 \text{ if } c^{\mathcal{I}^+} \neq 0\}.$$

**Example 11.4.3** Suppose  $h_+(c)$  is given by (11.4.2) with  $\|\cdot\| = \|\cdot\|_{\infty}$ . Then

$$\partial h_+(c) = \left\{ y \mid y_i = \begin{cases} \geq 0 & \text{if } c_i = h_+(c), \\ \leq 0 & \text{if } c_i = -h_+(c) \neq 0 \text{ and } i \in \mathcal{E}, \\ = 0 & \text{if } |c_i| < h_+(c) \neq 0 \text{ and } i \in \mathcal{E}, \\ & \text{or } c_i < h_+(c) \text{ and } i \in \mathcal{I}, \end{cases} \right. \\ \text{and } \sum_{i=1}^m |y_i| \left\{ \begin{array}{ll} = 1 & \text{if } h_+(c) > 0 \\ \leq 1 & \text{if } h_+(c) = 0 \end{array} \right\}.$$

Of course, the generalized gradients for the important special cases  $h_+(c) = \|c\|$  and  $h_+(c) = \|c^+\|$  for each of these norms may be recovered by setting  $\mathcal{I} = \emptyset$  and  $\mathcal{E} = \emptyset$ , respectively, in Examples 11.4.1–11.4.3.

It is trivial to obtain analogous results for the function (11.4.3), since we have the identity

$$h_-(c) = h_+(-c) \tag{11.4.11}$$

and thus

$$\partial h_-(c) = -\partial h_+(-c) \tag{11.4.12}$$

when  $\|\cdot\|$  is monotonic. We then have the following variant of Theorem 11.4.1.

**Corollary 11.4.2** If  $\|\cdot\|$  is monotonic and  $c$  is partitioned as (11.4.1), the generalized gradient of the function (11.4.3) is given by

$$\partial h_-(c) = \left\{ y = \begin{pmatrix} y_{\mathcal{E}} \\ y_{\mathcal{I}} \end{pmatrix} \mid \langle y, c \rangle = h_-(c), \quad y_{\mathcal{I}} \leq 0 \text{ and } \|y\|_{\mathbb{D}} \leq 1 \right\},$$

where the subscript D indicates the vector dual norm (2.3.1) (p. 21). Furthermore, if  $y \in \partial h_-(c)$ , then if  $j \in \mathcal{I}$  and  $c_j > 0$ , it follows that  $y_j = 0$ , and if  $h_-(c) > 0$ , then  $\|y\|_D = 1$ .

**Proof.** The result follows immediately from the identities (11.4.11) and (11.4.12), and by replacing  $y$  by  $-y$  in the conclusions of Theorem 11.4.1.  $\square$

For the  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$  norms, Corollary 11.4.2 gives the following.

**Example 11.4.4** Suppose  $h_-(c)$  is given by (11.4.3) with  $\|\cdot\| = \|\cdot\|_1$ . Then

$$\partial h_-(c) = \left\{ y \mid y_i = \begin{cases} 1 & \text{if } c_i > 0 \text{ and } i \in \mathcal{E}, \\ 0 & \text{if } c_i > 0 \text{ and } i \in \mathcal{I}, \\ -1 & \text{if } c_i < 0, \\ \in [-1, 1] & \text{if } c_i = 0 \text{ and } i \in \mathcal{E}, \\ \in [-1, 0] & \text{if } c_i = 0 \text{ and } i \in \mathcal{I}. \end{cases} \right\}$$

**Example 11.4.5** Suppose  $h_-(c)$  is given by (11.4.3) with  $\|\cdot\| = \|\cdot\|_2$ . Then

$$\partial h_-(c) = \{y \mid \|y\|_2 \leq 1, \quad y_{\mathcal{I}} \leq 0 \quad \text{and} \quad y = c^{\mathcal{I}^-} / \|c^{\mathcal{I}^-}\|_2 \quad \text{if } c^{\mathcal{I}^-} \neq 0\}.$$

**Example 11.4.6** Suppose  $h_-(c)$  is given by (11.4.3) with  $\|\cdot\| = \|\cdot\|_\infty$ . Then

$$\partial h_-(c) = \left\{ y \mid y_i \begin{cases} \geq 0 & \text{if } c_i = h_-(c) \neq 0 \text{ and } i \in \mathcal{E}, \\ \leq 0 & \text{if } c_i = -h_-(c), \\ = 0 & \text{if } |c_i| < h_-(c) \neq 0 \text{ and } i \in \mathcal{E}, \\ & \text{or } c_i > -h_-(c) \text{ and } i \in \mathcal{I}, \end{cases} \right. \\ \left. \text{and} \quad \sum_{i=1}^m |y_i| \begin{cases} = 1 & \text{if } h_-(c) > 0 \\ \leq 1 & \text{if } h_-(c) = 0 \end{cases} \right\}.$$

Now we turn to polyhedral convex functions of the form (11.4.4). We denote the index set of active functions at  $c$  by

$$\mathcal{A}(c) = \{i \mid \langle p_i, c \rangle + b_i = h_p(c) \text{ and } 1 \leq i \leq \ell\}.$$

Furthermore, given a set of vectors  $\{p_i\}$ ,  $i \in \mathcal{I}$ , we define

$$\text{co}\{p_i\}_{i \in \mathcal{I}} = \left\{ a \mid a = \sum_{i \in \mathcal{I}} \theta_i p_i, \text{ where } \theta_i \in [0, 1], \text{ and } \sum_{i \in \mathcal{I}} \theta_i = 1 \right\}$$

as the convex hull generated by the  $\{p_i\}$ . Given these definitions, we have the following classification.

**Theorem 11.4.3** The generalized gradient of the function

$$h_p(c) = \max_{1 \leq i \leq \ell} \langle p_i, c \rangle + b_i$$

is given by

$$\partial h_p(c) = \text{co}\{p_i\}_{i \in \mathcal{A}(c)}. \quad (11.4.13)$$

**Proof.** By definition, if  $\theta \in [0, 1]$ ,

$$\max_{1 \leq i \leq \ell} \langle p_i, c + (1 - \theta)d \rangle + b_i \leq \max_{1 \leq i \leq \ell} \langle p_i, c \rangle + b_i + (1 - \theta) \max_{1 \leq i \leq \ell} \langle p_i, d \rangle + b_i,$$

and thus  $h$  is a convex function of  $c$ . Hence (3.1.3) (p. 33) gives that

$$\partial h_p(c) = \{y \in \mathbb{R}^n \mid h_p(c) + \langle y, d \rangle \leq h_p(c + d) \text{ for all } d \in \mathbb{R}^n\}. \quad (11.4.14)$$

Suppose that there is a  $y$  in the set (11.4.14) that is not a member of the closed, convex set (11.4.13). Then the separating hyperplane theorem implies that there is a hyperplane  $\langle a, x \rangle = b$  such that  $\langle a, y \rangle > b$  and  $\langle a, x \rangle < b$ , for all  $x$  in the set (11.4.13), and thus that  $\langle a, y \rangle > \langle a, x \rangle$  for all such  $x$ . In particular,

$$\langle a, y \rangle > \langle a, p_i \rangle \text{ for } i \in \mathcal{A}(c), \quad (11.4.15)$$

by definition of the set (11.4.13). Now let  $d = \alpha a$  for some scalar  $\alpha$ . Then

$$\begin{aligned} h_p(c) + \langle y, d \rangle &= \max_{1 \leq i \leq \ell} \langle p_i, c \rangle + b_i + \alpha \langle y, a \rangle \\ &> \langle p_i, c \rangle + b_i + \alpha \langle p_i, a \rangle \quad \text{for } i \in \mathcal{A}(c), \text{ from (11.4.15)} \\ &= \max_{i \in \mathcal{A}(c)} \langle p_i, c + \alpha a \rangle + b_i \\ &= \max_{1 \leq i \leq \ell} \langle p_i, c + \alpha a \rangle + b_i \quad \text{for all } \alpha \text{ sufficiently small} \\ &= h_p(c + d), \quad \text{so that } \mathcal{A}(c + \alpha a) \subseteq \mathcal{A}(c) \end{aligned}$$

which contradicts the assumption that  $y$  is in the set (11.4.14). Thus the set (11.4.14) is a subset of (11.4.13).

Now suppose that  $y$  is in the set (11.4.13). Then we have that

$$y = \sum_{i \in \mathcal{A}(c)} \theta_i p_i,$$

where  $\theta_i \in [0, 1]$  and  $\sum_{i \in \mathcal{A}(c)} \theta_i = 1$ . Hence, if we let  $j$  be the index in  $\mathcal{A}(c)$  for which  $\langle p_i, d \rangle$  is largest and recall that  $h_p(c) = \langle p_i, c \rangle + b_i$  for all  $i \in \mathcal{A}(c)$ , it follows that

$$\begin{aligned} h_p(c) + \langle y, d \rangle &= \max_{1 \leq i \leq \ell} \langle p_i, c \rangle + b_i + \sum_{i \in \mathcal{A}(c)} \theta_i \langle p_i, d \rangle \\ &\leq \max_{i \in \mathcal{A}(c)} \langle p_i, c \rangle + b_i + \max_{i \in \mathcal{A}(c)} \langle p_i, d \rangle \\ &= \langle p_j, c + d \rangle + b_j \\ &\leq h_p(c + d) \end{aligned}$$

for all  $d$ , and thus  $y$  in the set (11.4.14).  $\square$

While we may rederive the results given in Examples 11.4.1, 11.4.3, 11.4.4, and 11.4.6, using Theorem 11.4.3, we may also derive new results like the following.

**Example 11.4.7** Suppose  $h_p(c) = \max_{1 \leq k \leq m} c_k$ . Then

$$\partial h_p(c) = \left\{ y \mid y \geq 0, \quad y_i = 0 \text{ if } c_i < \max_{1 \leq k \leq m} c_k \text{ and } \sum_{i=1}^m y_i = 1 \right\}.$$

In order to compute  $\|g(x)\|$  in any of these cases, (3.1.6) (p. 34) implies that we need to solve the convex minimization problem

$$\underset{y \in \partial h(c)}{\text{minimize}} \quad \|\nabla_x f(x) + (\nabla_x c(x))^T y\|. \quad (11.4.16)$$

Clearly (11.4.16) may be expressed equivalently as

$$\underset{y \in \partial h(c)}{\text{minimize}} \quad \langle y, \nabla_x c(x)(\nabla_x c(x))^T y \rangle + 2\langle y, \nabla_x c(x)\nabla_x f(x) \rangle,$$

since  $\|\cdot\|$  is the  $\ell_2$  norm. For Examples 11.4.2 and 11.4.5, the solution is completely defined by the constraints when  $h(c)$  is nonzero. The constraints for Examples 11.4.1, 11.4.3, 11.4.4, 11.4.6, and 11.4.7 are highly structured and linear; the resulting problems are then convex quadratic programs involving at most one general linear constraint. In many cases, such problems are trivial, or almost trivial, to solve.

In practice, in view of AA.1n, all we actually require is a bound on (11.4.16). A suitable bound may be obtained in a number of ways. Firstly, one might use duality to stop short of optimality. Secondly, although (11.4.16) is expressed in the  $\ell_2$  norm, one can use the equivalence of norms (see Section 2.3.1) to obtain bounds by solving the problem in another, more convenient norm. For instance, if the  $\ell_1$  or  $\ell_\infty$  norms are chosen, the resulting problems are linear programs.

## Notes and References for Section 11.4

Much of the material in this section is based upon that in Fletcher (1987a, Chapter 14). There

are many efficient methods for solving linear and quadratic programs. (See, for example, Gill, Murray, and Wright, 1981, or Fletcher, 1987b.) Some (e.g., Dussault, Ferland, and Lemaire, 1986) are especially tailored for the simple structures encountered in this section.

## 11.5 Suitable Models

We now turn to the selection of a suitable model for the problem. Clearly, from the point of view of the algorithms discussed in Sections 11.1 and 11.3, it is assumptions AM.1n–AM.4n that are crucial. While it is difficult to suggest an appropriate general-purpose model that satisfies these assumptions, a number of useful suggestions have been made when the objective function is the composite function  $f(x) + h(c(x))$  we considered in the last section. Since  $f$  and  $c$  are smooth, it is reasonable to approximate them by first- or higher order Taylor approximants. So long as  $h$  has exploitable structure, in the sense of its generalized gradient being simple to manipulate, we shall show that the models

$$m(x, p, s) = f(x) + \langle \nabla_x f(x), s \rangle + h\left(c(x) + \nabla_x c(x)s\right), \quad (11.5.1)$$

$$\begin{aligned} m(x, p, s) &= f(x) + \langle s, \nabla_x f(x) \rangle + \frac{1}{2}\langle s, \nabla_{xx} f(x)s \rangle \\ &\quad + h\left(c(x) + \nabla_x c(x)s + \frac{1}{2} \sum_{i=1}^m s_i \nabla_{xx} c_i(x)s\right), \end{aligned} \quad (11.5.2)$$

and

$$m(x, p, s) = f(x) + \langle \nabla_x f(x), s \rangle + \frac{1}{2}\langle s, Bs \rangle + h\left(c(x) + \nabla_x c(x)s\right) \quad (11.5.3)$$

are all appropriate—the matrix  $B$  in (11.5.3) is any bounded symmetric matrix intended to represent the overall curvature of the objective function.

The model (11.5.1) may be regarded as a first-order approximation to the objective function. If  $h$  is a polyhedral convex function of the form (11.4.4) (as we have seen, this includes problems based on the  $\ell_1$  or  $\ell_\infty$  norms) and if an  $\ell_1$ - or  $\ell_\infty$ -norm trust region is used, the trust-region problem may be expressed as a linear program. To see this, consider, for example, (11.4.4) with an  $\ell_\infty$ -norm trust region. In this case, the trust-region subproblem (minimize the model within the trust region  $\|s\| \leq \Delta$ ) is

$$\begin{aligned} \text{minimize}_{s \in \mathbb{R}^n} \quad & \langle \nabla_x f(x), s \rangle + \max_{1 \leq i \leq \ell} \langle p_i, c(x) + \nabla_x c(x)s \rangle + b_i \\ \text{subject to} \quad & -\Delta e \leq s \leq \Delta e, \end{aligned} \quad (11.5.4)$$

where we have neglected the constant term  $f(x)$ . If we define an artificial variable

$$t = \max_{1 \leq i \leq \ell} \langle p_i, c(x) + \nabla_x c(x)s \rangle + b_i,$$

(11.5.4) is equivalent to the linear program

$$\begin{aligned} \text{minimize}_{s \in \mathbb{R}^n, t \in \mathbb{R}} \quad & \langle \nabla_x f(x), s \rangle + t \quad \text{subject to} \quad -\Delta e \leq s \leq \Delta e \\ \text{and} \quad & \langle (\nabla_x c(x))^T p_i, s \rangle + \langle p_i, c(x) \rangle + b_i \leq t \quad \text{for } 1 \leq i \leq \ell. \end{aligned} \quad (11.5.5)$$

When an  $\ell_1$ -norm trust region is used, a similar problem is obtained, except that the resulting trust-region constraint

$$\sum_{i=1}^n |s_i| \leq \Delta$$

is replaced by

$$\sum_{i=1}^n u_i + v_i \leq \Delta, \quad s_i = u_i - v_i \text{ for } 1 \leq i \leq \ell \quad \text{and } (u, v) \geq 0,$$

and the minimization is performed additionally over the variables  $u$  and  $v$ . In both cases, the structure resulting from the extra artificial variables  $t$ ,  $u$ , and  $v$  may be exploited.

While it is possible to derive a globally convergent method based on the model (11.5.2), it is unlikely that such a method will be capable of sustaining a fast rate of convergence. To overcome this deficiency, the model (11.5.2) may be considered as a second-order approximation to  $f(x+s)+h(c(x+s))$ . The disadvantage of this model is that the trust-region subproblems are no longer necessarily easy to solve. For example, if  $h$  is a polyhedral convex function, the resulting subproblem involves the minimization of a (possibly nonconvex) quadratic objective subject to a set of (possibly nonconvex) quadratic inequalities, in addition to whatever constraints are imposed from the trust region.

Thus we may prefer the compromise model (11.5.3), which allows second-order information, but simply adds a quadratic term  $\frac{1}{2}\langle s, Bs \rangle$  to the trust-region objective function. As a consequence, if  $h$  is polyhedral and convex, and if a polyhedral trust region is used, the trust-region subproblem is equivalent to a (possibly nonconvex) quadratic program; for example, with an  $\ell_\infty$ -norm trust-region, we simply add the term  $\frac{1}{2}\langle s, Bs \rangle$  to the objective function for the model problem (11.5.5). Since there are effective methods for finding (at least) local solutions to such problems, the model (11.5.3) offers a considerable advantage over (11.5.2). However, a natural question is what we should choose for  $B$ . To answer this, we return to (11.5.2) and consider what conditions must be satisfied if we are to succeed in minimizing this model (in the absence of a trust region). In view of Corollary 3.2.13 (p. 48), there will be some

$$y \in \partial h \left( c(x) + \nabla_x c(x)s + \frac{1}{2} \sum_{i=1}^m s_i \nabla_{xx} c_i(x)s \right) \quad (11.5.6)$$

for which

$$\nabla_x f(x) + \nabla_{xx} f(x)s + (\nabla_x c(x))^T y + \sum_{i=1}^m y_i \nabla_{xx} c_i(x)s = 0.$$

We may rearrange this equation to get

$$\nabla_x f(x) + Bs + (\nabla_x c(x))^T y = 0,$$

where

$$B = \nabla_{xx} f(x) + \sum_{i=1}^m y_i \nabla_{xx} c_i(x). \quad (11.5.7)$$

Now compare this with the conditions

$$\nabla_x f(x) + \nabla_{xx} f(x)s + (\nabla_x c(x))^T y + \sum_{i=1}^m y_i \nabla_{xx} c_i(x)s = 0$$

for some

$$y \in \partial h \left( c(x) + \nabla_x c(x)s \right), \quad (11.5.8)$$

which must be satisfied if we succeed in minimizing the model (11.5.3) (in the absence of a trust region). Then the two problems have the same solution so long as the sets on the right-hand sides of (11.5.6) and (11.5.8) are the same, and it may be reasonable to suppose that the solutions are close if  $s$  is small. Thus, it is common to pick  $B$  of the form (11.5.7), where the parameters  $y$  are chosen, for example, as the optimal generalized gradient of  $h$  at the solution of a previous trust-region subproblem. Most importantly, since  $h$  is regular and locally Lipschitz, the set of generalized gradients is bounded, and thus this choice of  $B$  will be bounded so long as the second derivatives of  $f$  and  $c$  stay bounded.

It remains for us to show that the models (11.5.1)–(11.5.3) really are suitable for our purpose, namely that they satisfy AM.1n–AM.4n. Provided  $f$  and  $c$  are smooth and  $h$  is locally Lipschitz and regular, this is so, as we now show.

**Theorem 11.5.1** Suppose that  $h$  is a regular, locally Lipschitz continuous function on a subset  $\mathcal{C}$  of  $\mathbb{R}^m$  and that  $f$  and  $c$  are continuously differentiable functions, with Lipschitz continuous derivatives, from a compact subset  $\mathcal{X} \subset \mathbb{R}^n$  to  $\mathbb{R}$  and  $\mathcal{C}$ , respectively. Then the model (11.5.1) satisfies AM.1n–AM.4n on  $\mathcal{X}$ . If  $f$  and  $c$  are twice-continuously differentiable, the same is true of the model (11.5.2). Finally, if  $B$  is bounded, the same result follows for the model (11.5.3).

**Proof.** Since all four models are formed as the sums of continuously differentiable functions and regular and locally Lipschitz functions of continuously differentiable functions, AM.1n holds. As the first two models do not depend on parameters, while the latter two simply depend on the bounded matrix  $B$  and vector  $s^E$ , AM.4n holds. Thus it remains to show that AM.2n and AM.3n hold. Since it follows from Lemma 11.2.1 that this is equivalent to showing that (11.2.1) holds, we choose to consider the latter.

First consider the model (11.5.1). We have that

$$\begin{aligned} \theta(x, p, s) &= \frac{f(x + s) + h(c(x + s)) - m(x, p, s)}{\|s\|} \\ &= \frac{f(x + s) - f(x) - \langle \nabla_x f(x), s \rangle + h(c(x + s)) - h \left( c(x) + \nabla_x c(x)s \right)}{\|s\|}. \end{aligned}$$

Dividing this expression into two parts, and examining each part separately, we

have, from Theorem 3.1.4 (p. 29), that

$$\frac{|f(x+s) - f(x) - \langle \nabla_x f(x), s \rangle|}{\|s\|} \leq \frac{1}{2}\gamma_1\|s\|$$

for the first, and from the local Lipschitz continuity of  $h$  and Theorem 3.1.6 (p. 29), that

$$\frac{|h(c(x+s)) - h(c(x) + \nabla_x c(x)s)|}{\|s\|} \leq \gamma_2 \frac{\|c(x+s) - c(x) - \nabla_x c(x)\|}{\|s\|} \leq \frac{1}{2}\gamma_2\gamma_3\|s\|$$

for the second, where  $\gamma_1$ ,  $\gamma_2$ , and  $\gamma_3$  are appropriate Lipschitz constants. Thus  $\theta(x, p, s)$  converges to zero with  $\|s\|$ , and hence Lemma 11.2.1 ensures that AM.2n and AM.3n hold for (11.5.1).

The proof that AM.2n and AM.3n hold for the models (11.5.2) and (11.5.3) is essentially the same, there simply being extra terms of  $O(\|s\|)$  in  $\theta(x, p, s)$  which result from the second-order terms in the definitions of (11.5.2) and (11.5.3).  $\square$

## Notes and References for Section 11.5

The minimization of polyhedral convex functions over systems of linear constraints was considered in considerable detail by Osborne (1985). An algorithm for the special but important case of minimization within an  $\ell_\infty$  trust region was given by Xu and Zhang (1995), while the problem of minimizing certain classes of piecewise quadratic functions like (11.5.2) was considered by Sun (1997).

Although, as we have said, such problems can often be transformed into linear or quadratic programs and solved as such (using, for example, any of the methods we shall survey in Section 15.2.3), it is wise in practice to exploit the structure due to the transformation. There are three ways in which this is possible. Firstly, the algebraic structure due to the artificial variables is extremely regular and may be exploited within many of the matrix factorizations that are used to determine search directions within linear and quadratic program algorithms. Secondly, once a search direction has been chosen, it is easy to find the exact minimizer of the model in this direction, since the model is a univariate piecewise linear or quadratic function along any line; this may involve passing “through” one or more piecewise segments before the required line-minimizer is discovered. (see the related calculation in Section 17.3). Finally, it is always trivial to pick the values of the artificial variables to ensure feasibility of the constraints. See Conn and Sinclair (1975) and Fletcher (1985) for more details.

The models given here, as well as others, are considered by Dennis, Li, and Tapia (1995). As they show, the algorithm of Section 11.1 and the models considered in this section are directly applicable to a variety of problems and may be used to unify the convergence results for a number of existing algorithms. In particular, they apply to appropriate forms of Madsen (1975), Powell (1983), and Yuan’s (1983) methods for solving nonlinear fitting problems, and Duff, Nocedal, and Reid (1987) and El-Hallabi and Tapia’s (1993) methods for solving systems of nonlinear equations, as well as Zhang, Kim, and Lasdon (1985) and Fletcher’s (1987a) methods for solving constrained optimization problems (see Chapter 15). However, some of these algorithms are known to converge under weaker assumptions than required in this chapter. In particular, Yuan (1985a) shows that his method for solving nonlinear fitting problems

using a model of the form (11.5.3) converges under weaker conditions of the form AM.4d on the required Hessian approximations.

Bell (1990) and Terpolilli (1995) considers variants in which the function values are not assumed to be exact. Methods for problems including constraints are considered by Gabriel and Pang (1994), Jiang and Qi (1996), and Martínez and Moretti (1997), while the solution of nonsmooth sets of nonlinear equations is investigated by Qi (1995).

## Part III

---

# Trust-Region Methods for Constrained Optimization with Convex Constraints

In this part of the book, we consider methods that are relevant when the constraint set defines a convex region. This includes simple bounds on the variables, linear constraints, and a variety of other constraint sets defined by simple geometries.

---

## Chapter 12

---

# Projection Methods for Convex Constraints

---

Having examined the unconstrained optimization problem (6.1.1) (p. 115) and its variants, we now turn to the frequent case where the formulation of the problem includes restrictions on the variables. In other words, we now consider the problem

$$\underset{x \in \mathcal{C}}{\text{minimize}} \quad f(x), \quad (12.0.1)$$

where the set  $\mathcal{C} \subset \mathbb{R}^n$  is the *feasible region*, that is, the set of vectors of variables among which a best solution is sought. For instance, one may wish to find the settings for optimal performance of some machine whose operational parameters must lie in certain prescribed intervals (pressures, surfaces, electric resistances, pipe diameters, or populations cannot be negative!). Or we may seek to determine at which rate dangerous drugs can be injected into patients without killing them by overdose. In fact, the situation where  $\mathcal{C}$  is a proper subset of  $\mathbb{R}^n$  is extremely common in applications, and many of these simply do not make any sense unless restrictions on the variables are included.

Of course, the nature of the set  $\mathcal{C}$  has a major influence on the methods we will consider for solving our problem. Despite their importance, in this book we will not consider feasible regions that are not connected (such as  $\mathcal{C} = \mathbb{N}^n$  or  $\{0, 1\}^n$ ), and we will concentrate on connected feasible sets that can be described by a finite set of smooth equality or inequality constraints. Exactly what assumptions we will make on  $\mathcal{C}$  will be stated explicitly, and discussed in detail, as each different context arises.

### 12.1 Simple Feasible Domains and Their Geometry

In this chapter, we will restrict ourselves to the case where the feasible region is reasonably simple in the sense that  $\mathcal{C}$  is a nonempty closed *convex* subset of  $\mathbb{R}^n$  (see AC.2) on which *projections* are relatively easy to calculate. The purpose of this section is to

explore what types of feasible regions are covered by this rather broad definition, to deduce what we can say about their geometries, and to discuss basic properties of their associated projection operators.

### 12.1.1 Simple Bounds on the Variables

One of the simplest feasible regions  $\mathcal{C}$  is obtained when the only restrictions on the problem's variables are that they must lie in known intervals; that is,

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid x_\ell \leq x \leq x_u\}, \quad (12.1.1)$$

where the inequalities are meant componentwise (meaning that  $[x_\ell]_i \leq [x]_i \leq [x_u]_i$  for  $i = 1, \dots, n$ ). The vectors  $x_\ell$  and  $x_u$  are the vectors of *lower* and *upper bounds* on the variables, respectively, and we allow either of them to have components  $-\infty$  or  $+\infty$ . Despite their simplicity, bound constraints are very important in practice. They arise in many applications in their own right, but also appear in the solution of subproblems in more complicated situations. For instance, we have already seen that we could describe the trust-region subproblem using the  $\ell_\infty$  norm (see Section 6.7) as a problem of this type.

One very useful property of boxes of the form (12.1.1) is that the projection of any vector  $y$  onto  $\mathcal{C}$  is extremely easy to calculate. Indeed, we see immediately that this projection, which we shall write as  $P_{\mathcal{C}}$  (or simply  $P$  if no confusion is possible), is given componentwise by

$$[P_{\mathcal{C}}(y)]_i \stackrel{\text{def}}{=} \begin{cases} [x_\ell]_i & \text{if } [y]_i \leq [x_\ell]_i, \\ [y]_i & \text{if } [x_\ell]_i < [y]_i < [x_u]_i, \\ [x_u]_i & \text{if } [x_u]_i \leq [y]_i \end{cases} \quad (12.1.2)$$

for  $i = 1, \dots, n$ . (Figure 12.1.1 illustrates this definition.) As a consequence, we may certainly say that *applying  $P_{\mathcal{C}}$  is very cheap when  $\mathcal{C}$  is a box*. We will therefore construct trust-region algorithms that attempt to exploit this feature.

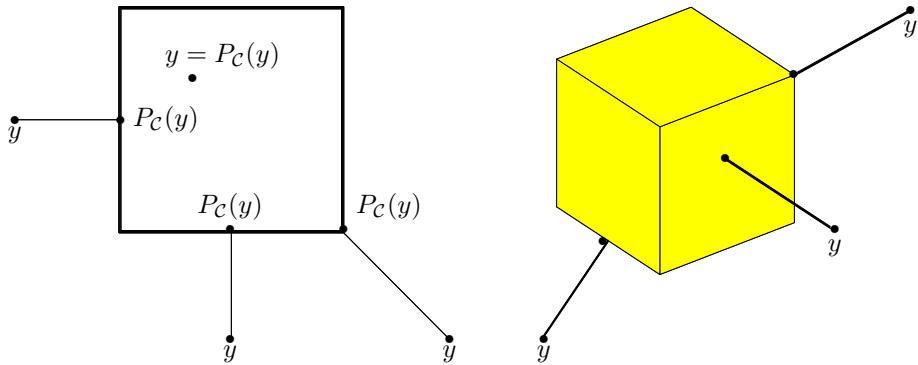


Figure 12.1.1: Projection onto a box in two and three dimensions.

Finally, bound constraints are also appealing because their normals (the vectors  $e_i$  of the canonical basis) are by construction linearly independent, which means they necessarily satisfy the constraint qualification.

### 12.1.2 Other Simple Domains

There are other domains on which projection is a relatively easy task to perform. For instance, projections onto a sphere or a cylinder are trivial, as we illustrate in Figures 12.1.2 and 12.1.3. To be specific, if

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid \|x - x_0\|_2 \leq \delta\},$$

then

$$P_{\mathcal{C}}(y) = x_0 + \delta \frac{y - x_0}{\|y - x_0\|_2}. \quad (12.1.3)$$

The projection onto a cylinder is similar, as it reduces to the projection onto a sphere in the subspace orthogonal to the cylinder generating directions.

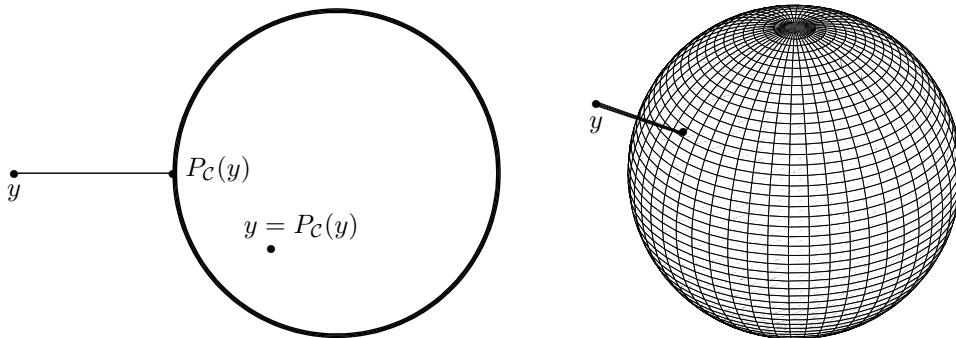


Figure 12.1.2: Projection onto a sphere in two and three dimensions.

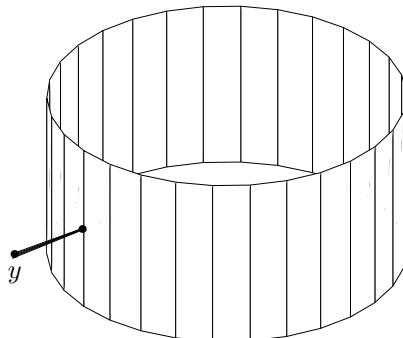


Figure 12.1.3: Projection onto a cylinder.

Another domain on which projection is relatively easy is the *order-simplex*, defined by

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq x_n\}.$$

This is of interest, for instance, when the variables  $x_i$  represent the values of a non-decreasing one-dimensional functional at certain specific points (nodes). The shape of the order-simplex in  $\mathbb{R}^3$  is shown in the shaded part of Figure 12.1.4.

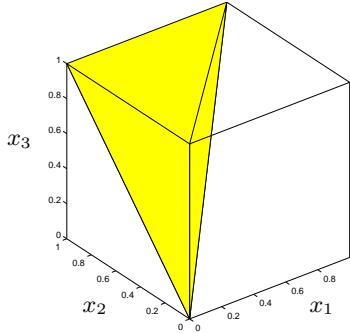


Figure 12.1.4: The shape of the order-simplex in  $\mathbb{R}^3$ .

A very nice algorithm for calculating the projection onto the order-simplex was proposed by Best and Chakravarti (1990), but we do not describe it in detail here. The crux of this technique is that the projection of a vector  $y \in \mathbb{R}^n$  onto  $\mathcal{C}$  may be determined in  $O(n)$  operations, as is also the case for the projections (12.1.2) and (12.1.3).

### 12.1.3 The Projected-Gradient Path

We know from Theorem 3.2.9 (p. 45), that a point  $x_*$  is first-order critical for problem (12.0.1) if the gradient of the objective function at  $x_*$  belongs to the normal cone  $\mathcal{N}(x_*)$ . This condition may be interpreted as the requirement that the projection of any point along the Cauchy arc from  $x_*$  is  $x_*$  itself. Conversely, if  $x$  is not first-order critical, we may be interested in analysing properties of projections of points along the Cauchy arc from  $x$  as a constructive way to determine better points. We thus define, for any vector  $x$  belonging to  $\mathcal{C}$ , the *projected-gradient path* as

$$p(t, x) = P_{\mathcal{C}}[x - t\nabla_x f(x)] \quad (12.1.4)$$

for all  $t \geq 0$ . We illustrate this definition in Figure 12.1.5.

As this figure indicates, it may happen that the projected-gradient path “stops”, in the sense that all values of the parameter  $t$  exceeding a threshold value  $t_m$  produce the same projected vector  $p(t_m, x)$ . In fact, this situation can be characterized in terms of the normal cone at  $p(t_m, x)$  as follows.

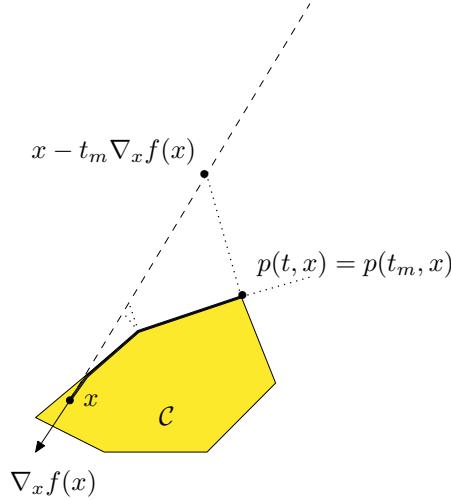


Figure 12.1.5: The projected-gradient path.

**Theorem 12.1.1** Suppose that AC.2 and AO.1 hold. Suppose furthermore that, for some  $x$  and some  $t_m \geq 0$ ,

$$-\nabla_x f(x) \in \mathcal{N}(p(t_m, x)).$$

Then

$$p(t, x) = p(t_m, x) \text{ for all } t \geq t_m.$$

**Proof.** Let  $z = p(t_m, x)$ . We then obtain from the definition of  $p(t, x)$  and (3.1.8) (p. 36) that, for  $t \geq t_m$ ,

$$\begin{aligned} p(t, x) &= P_C [x - t \nabla_x f(x)] \\ &= P_C [x - t_m \nabla_x f(x) - (t - t_m) \nabla_x f(x)] \\ &= P_C [x - t_m \nabla_x f(x)] \\ &= p(t_m, x) \end{aligned}$$

because  $-(t - t_m) \nabla_x f(x) \in \mathcal{N}(p(t_m, x))$ .  $\square$

We next prove that first-order criticality can easily be expressed in a form that uses the projected-gradient path.

**Theorem 12.1.2** Suppose that AC.2 and AO.1 hold. Then the point  $x_* \in \mathcal{C}$  is a first-order critical point for problem (12.0.1) if and only if

$$p(t, x_*) = x_* \text{ for all } t \geq 0.$$

**Proof.** This is an immediate consequence of Theorems 3.2.9 (p. 45) and 12.1.1.  $\square$

We continue by reviewing some useful properties of the projected-gradient path.

**Theorem 12.1.3** Suppose that AF.1 and AC.2 hold and that  $x \in \mathcal{C}$ . Then the function

$$\phi(t) = \|p(t, x) - x\|$$

is nondecreasing for  $t \geq 0$ . If AC.1 and AC.7 also hold, the limit

$$\lim_{t \rightarrow \infty} \phi(t) < \infty \quad (12.1.5)$$

implies that

$$\lim_{t \rightarrow \infty} \|P_{T(p(t, x))}[-\nabla_x f(x)]\| = 0, \quad (12.1.6)$$

where  $T(x)$  is the tangent cone at  $x$  with respect to  $\mathcal{C}$ .

**Proof.** The proof of the first statement immediately follows from the monotone nature of the projection operator. To prove the second statement, consider the sequence of points whose  $i$ th element is  $x_i \stackrel{\text{def}}{=} p(i\epsilon, x)$ , where we choose  $\epsilon > 0$  small enough to ensure the inequality

$$\epsilon \delta_1 \|\nabla_x f(x)\| \leq \frac{1}{2},$$

and where

$$\delta_1 = \max_{i=1,\dots,m} \max_{\|\xi-x\| \leq \delta_0} \|\nabla_{xx} c_i(\xi)\|$$

for some  $\delta_0 \geq \lim_{t \rightarrow \infty} \phi(t)$ , and the  $c_i$  are the constraint functions implied by AC.7. Because of the first statement and (12.1.5), we have that  $\|x_i - x\|$  converges to some finite value  $\delta_2 > 0$ . This means that, for  $i$  sufficiently large,  $x_i$  converges to the projection of the point  $x - i\epsilon \nabla_x f(x)$  onto a sphere of radius  $\delta_2$  centered at  $x$ . But the limit of these projections is unique, which implies that the sequence  $\{x_i\}$  also converges to a finite limit  $x_*$ . Thus  $\|x_{i+1} - x_i\|$  converges to zero, which implies that  $\mathcal{A}(x_i) = \mathcal{A}_\infty(x)$ , for some<sup>185</sup> active set  $\mathcal{A}_\infty(x)$ , because AC.7 guarantees that there are only finitely many faces in  $\mathcal{C}$ . As a consequence,  $x_{i+1}$  is the projection of  $z_i = x_i + \epsilon P_{T(x_i)}[-\nabla_x f(x)]$  onto the surface of the constraints that are active at  $x_i$ , and

$$\|z_i - x_i\| = \epsilon \|P_{T(x_i)}[-\nabla_x f(x)]\| \leq \epsilon \|\nabla_x f(x)\| \leq \frac{1}{2\delta_1}, \quad (12.1.7)$$

because of the definition of  $\epsilon$ . Furthermore, AC.1 and a Taylor's expansion of the constraints imply that

$$\|x_{i+1} - z_i\| \leq \|x_{i+1} - \tilde{x}_{i+1}\| \leq \delta_1 \|z_i - x_i\|^2,$$

<sup>185</sup>Note that  $\mathcal{A}(x_*)$  may be larger than  $\mathcal{A}_\infty(x)$ , as can be checked by considering the two-dimensional problem  $\max x_2$  subject to  $(x_1 - 1)^2 + x_2^2 \leq 1$  and  $x_1 \leq 1$  at the point  $x = (0, 0)$ . In this case,  $\mathcal{A}_\infty(x)$  only contains the quadratic constraint, while both constraints are active at  $x_* = (1, 1)$ .

where  $\tilde{x}_{i+1}$  is the intersection of the constraint boundary with the line perpendicular to  $z_i - x_i$  in the plane containing  $x_i$ ,  $x_{i+1}$ , and  $z_i$ . Thus

$$\begin{aligned}\|x_{i+1} - x_i\| &\geq \|z_i - x_i\| - \|x_{i+1} - z_i\| \\ &\geq \|z_i - x_i\| - \delta_1 \|z_i - x_i\|^2 \\ &\geq \frac{1}{2} \|z_i - x_i\| \\ &= \frac{1}{2} \epsilon \|P_{\mathcal{T}(x_i)}[-\nabla_x f(x)]\|,\end{aligned}$$

where we used (12.1.7). This last bound and the fact that  $\|x_{i+1} - x_i\|$  tends to zero when  $i$  tends to infinity then give (12.1.6), as desired.  $\square$

To conclude this brief analysis of the properties of the projected-gradient path, we show the important property that each point on this path is the solution of a specific (constrained) optimization problem.

**Theorem 12.1.4** Suppose that AF.1 and AC.2 hold and that  $x \in \mathcal{C}$ . Then, for each point  $p(t, x)$  on the projected-gradient path (12.1.4),  $p(t, x) - x$  is a solution of the problem

$$\min_{\substack{x+d \in \mathcal{C} \\ \|d\| \leq \theta}} \langle \nabla_x f(x), d \rangle, \quad (12.1.8)$$

where  $\theta = \|p(t, x) - x\|$ . Furthermore,  $p(t, x) - x$  is also a solution of problem (12.1.8) for all  $\theta \geq \|p(t, x) - x\|$  whenever

$$-\nabla_x f(x) \in \mathcal{N}(p(t, x)). \quad (12.1.9)$$

**Proof.** Consider the problem of projecting  $x - t\nabla_x f(x)$  onto  $\mathcal{C}$ , that is,

$$\min_{x+u \in \mathcal{C}} \|x - t\nabla_x f(x) - x - u\|_2^2 = \min_{x+u \in \mathcal{C}} \|t\nabla_x f(x) + u\|_2^2. \quad (12.1.10)$$

Suppose first that  $t = 0$ . Then  $p(t, x) = x$  and the solution of problem (12.1.10) is  $u = 0$ . In this case, since  $\theta = p(t, x) - x = 0$ ,  $d = 0$  is a solution of problem (12.1.8). Suppose now that  $t > 0$  and that  $u$  solves (12.1.10). Note that the first-order optimality conditions for this problem are

$$-t\nabla_x f(x) - u \in \mathcal{N}(x + u)$$

(see Theorem 3.2.11 [p. 46]). Since  $t > 0$  and  $\mathcal{N}(x + u)$  is a cone, this condition may be rewritten as

$$-\frac{1}{t}u - \nabla_x f(x) \in \mathcal{N}(x + u). \quad (12.1.11)$$

On the other hand, problem (12.1.8) is equivalent to

$$\min_{\substack{x+d \in \mathcal{C} \\ \|d\|^2 \leq \theta^2}} \langle \nabla_x f(x), d \rangle,$$

whose first-order optimality conditions are given by

$$-2\lambda d - \nabla_x f(x) \in \mathcal{N}(x + d), \quad \lambda \geq 0 \text{ and } \lambda(\|d\|^2 - \theta^2) = 0. \quad (12.1.12)$$

The first of these conditions is then identical to (12.1.11) if one sets

$$d = u \text{ and } \lambda = \frac{1}{2t} > 0,$$

and the second is also satisfied. The third immediately results from the identity

$$\theta = \|d\| \quad (= \|u\| = \|p(t, x) - x\|).$$

Suppose now that (12.1.9) holds. It is immediate to verify that  $d = p(t, x) - x$  and  $\lambda = 0$  then satisfy conditions (12.1.12) for all  $\theta \geq \|p(t, x) - x\|$ , which concludes the proof.  $\square$

#### 12.1.4 A New Criticality Measure

We now consider the definition of a new criticality measure based on the notion of the gradient path. We start by defining, for  $x \in \mathcal{C}$ ,

$$\chi(x, \theta) \stackrel{\text{def}}{=} \left| \min_{\substack{x+d \in \mathcal{C} \\ \|d\| \leq \theta}} \langle \nabla_x f(x), d \rangle \right|. \quad (12.1.13)$$

Notice that this is minus the optimal value of problem (12.1.8). We then have the following properties.

**Theorem 12.1.5** Suppose that AF.1 and AC.2 hold and that  $x \in \mathcal{C}$ . Then,

- (i) the function  $\chi(x, \theta)$  is continuous and nondecreasing as a function of  $\theta$  for all  $\theta \geq 0$ ,
- (ii) the function  $\frac{\chi(x, \theta)}{\theta}$  is nonincreasing as a function of  $\theta$  for all  $\theta > 0$ ,
- (iii) for any  $d$  such that  $x + d \in \mathcal{C}$ , the inequality

$$\chi(x, \theta) \leq |\langle \nabla_x f(x), d \rangle| + 2\theta \|P_{\mathcal{T}(x+d)}[-\nabla_x f(x)]\|$$

holds for all  $\theta > \|d\|$ ,

- (iv) the inequality

$$\frac{\chi(x, \theta)}{\theta} \leq \|P_{\mathcal{T}(x)}[-\nabla_x f(x)]\|$$

holds for all  $\theta > 0$ .

**Proof.** The first statement is an immediate consequence of the definition (12.1.13) and the continuity of the solution of convex optimization problems (see Theorem 3.2.8 [p. 44]). In order to prove the second statement, consider  $0 < \theta_1 < \theta_2$  and two vectors  $d_1$  and  $d_2$  such that

$$\chi(x, \theta_j) = -\langle \nabla_x f(x), d_j \rangle, \quad \|d_j\| \leq \theta_j, \quad x + d_j \in \mathcal{C}, \quad j = 1, 2.$$

We observe that the point  $x + (\theta_1/\theta_2)d_2$  belongs to the segment  $[x, x + d_2]$  and therefore to  $\mathcal{C}$ , because of AC.2. Furthermore,

$$\left\| \frac{\theta_1}{\theta_2} d_2 \right\| = \frac{\theta_1}{\theta_2} \|d_2\| \leq \theta_1$$

and the point  $x + (\theta_1/\theta_2)d_2$  thus lies in the feasible domain of the optimization problem associated with the definition of  $\chi(x, \theta_1)$  and  $d_1$ . As a consequence, we have that

$$\frac{\chi(x, \theta_1)}{\theta_1} \geq \frac{1}{\theta_1} \left| \left\langle \nabla_x f(x), \frac{\theta_1}{\theta_2} d_2 \right\rangle \right| = \frac{\chi(x, \theta_2)}{\theta_2},$$

and the second statement is proved.

The third statement is proved as follows. Let  $d_*$  be a solution of problem (12.1.8) (which implies that  $\|d_*\| \leq \theta$ ). Then, from (12.1.13) and the triangle inequality, we have that

$$\chi(x, \theta) \leq |\langle \nabla_x f(x), d \rangle| + |\langle \nabla_x f(x), d_* - d \rangle|. \quad (12.1.14)$$

Moreover,

$$d_* - d = (x + d_*) - (x + d) \in \mathcal{T}(x + d), \quad (12.1.15)$$

since  $x + d_* \in \mathcal{C}$  by definition of  $d_*$ . Applying now the Moreau decomposition (3.1.7) (p. 36) to  $-\nabla_x f(x)$  at the point  $x + d$ , we obtain that

$$\begin{aligned} \langle \nabla_x f(x), d_* - d \rangle &= -\langle P_{\mathcal{T}(x+d)}[-\nabla_x f(x)], d_* - d \rangle - \langle P_{\mathcal{N}(x+d)}[-\nabla_x f(x)], d_* - d \rangle \\ &\geq -\langle P_{\mathcal{T}(x+d)}[-\nabla_x f(x)], d_* - d \rangle, \end{aligned}$$

where we used (12.1.15) and the fact that the tangent cone is the polar of the normal cone (see p. 35) to derive the last inequality. Taking absolute values and applying the Cauchy–Schwarz inequality thus yields that

$$|\langle \nabla_x f(x), d_* - d \rangle| \leq \|d_* - d\| \|P_{\mathcal{T}(x+d)}[-\nabla_x f(x)]\|. \quad (12.1.16)$$

Since  $\|d_* - d\| \leq \|d\| + \|d_*\| \leq 2\theta$ , substituting (12.1.16) in (12.1.14) gives the third statement of the theorem.

Finally, we obtain the fourth conclusion by choosing  $d = 0$  in (12.1.16), applying the definition of  $\chi(x, \theta)$  in (12.1.13) and recalling that  $\|d_*\| \leq \theta$ .  $\square$

We then observe that problem (12.1.8) may in turn be used to define a first-order criticality measure.

**Theorem 12.1.6** Suppose that AF.1 and AC.2 hold. Then the quantity

$$\chi(x) \stackrel{\text{def}}{=} \chi(x, 1) = \left| \min_{\substack{x+d \in \mathcal{C} \\ \|d\| \leq 1}} \langle \nabla_x f(x), d \rangle \right|, \quad (12.1.17)$$

defined for  $x \in \mathcal{C}$ , is a first-order criticality measure, in the sense of Section 8.1.1.

**Proof.** Recalling the definition of a first-order criticality measure on p. 249, we have to prove that  $\chi(x)$  is a nonnegative, continuous function of  $x$  that vanishes if and only if  $x$  is first-order critical. The continuity of  $\chi(x)$  results from Theorem 3.2.8 (p. 44) on the continuity of solutions of convex optimization problems. Its nonnegativity is a result of its definition. Furthermore, if  $\chi(x) = 0$ , Theorem 12.1.4 implies that  $p(t, x) = x$  for all  $t \geq 0$ , which then ensures that  $x$  is first-order critical because of Theorem 12.1.2. Conversely, the same theorem implies that, if  $x$  is first-order critical, then  $p(t, x) = x$  for all  $t \geq 0$ , and therefore  $\chi(x) = 0$ .  $\square$

Note that  $\chi(x)$  can be interpreted as the progress that can be made on a first-order model at  $x$  in a ball of radius unity (with the constraint of preserving feasibility, of course). In this sense, it is a direct generalization of  $\|\nabla_x f(x)\|$ , which can be interpreted in exactly the same way when the problem is unconstrained. In fact,

$$\chi(x) = \|\nabla_x f(x)\| \text{ whenever } \mathcal{C} = \mathbb{R}^n.$$

The function  $\chi(x)$  defined in (12.1.17) is not the only criticality measure possible for problem (12.0.1) when  $\mathcal{C}$  is convex. One possibility that immediately comes to mind is to consider

$$\|P_{\mathcal{T}(x)}[-\nabla_x f(x)]\|. \quad (12.1.18)$$

Unfortunately, the tangent cone  $\mathcal{T}(x)$  is a discontinuous function of  $x$ , which means that the value of (12.1.18) may not be small even arbitrarily close to a first-order critical point. In our opinion, this lack of continuity makes (12.1.18) difficult to interpret, and more importantly means that this choice is not actually a criticality measure. A far better alternative is to consider

$$\chi(x) = \|P_{\mathcal{C}}[x - \nabla_x f(x)] - x\|, \quad (12.1.19)$$

which plays a similar geometric role while maintaining continuity (and thus is a criticality measure). Figure 12.1.6 illustrates both  $P_{\mathcal{T}(x)}[-\nabla_x f(x)]$  and  $P_{\mathcal{C}}[x - \nabla_x f(x)] - x$  at three points in a convex feasible region. In this figure, these two vectors are represented with thick lines, while the negative gradients are shown as thin arrows, orthogonal to the contour lines of a linear  $f(x)$ . The dotted lines indicate the projections of  $x - \nabla_x f(x)$  onto the feasible region. One clearly sees in the top picture that  $\|P_{\mathcal{T}(x)}[-\nabla_x f(x)]\|$  has discontinuities on the boundary of the feasible region (its value remains large in the interior of the feasible set, arbitrarily close to the corner, where it vanishes), while the bottom picture demonstrates that  $\|P_{\mathcal{C}}[x - \nabla_x f(x)] - x\|$  is continuous.

## Notes and References for Section 12.1

The theory of projections on convex sets is a very rich and old subject. We refer the reader to Zarantonello (1971) for an in-depth survey of this area. The use of the projected-gradient path in the theory of trust-region methods was explicit in the paper by Conn, Gould, and Toint (1988a), where the criticality measure (12.1.19) was also introduced. The results stated

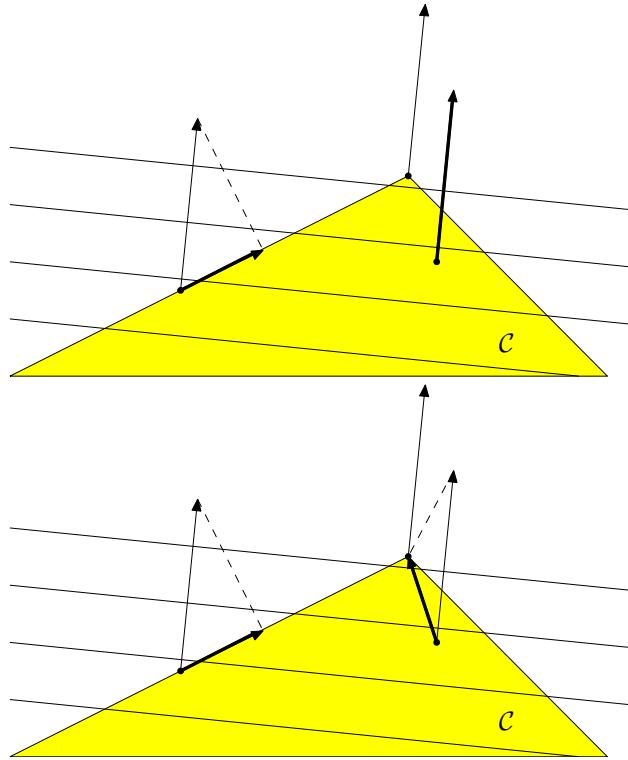


Figure 12.1.6: The projection of the negative gradient onto the tangent cone,  $P_{T(x)}[-\nabla_x f(x)]$  (top), and  $P_C[x - \nabla_x f(x)] - x$  (bottom).

in Theorem 12.1.5 have been part of the mathematical programming folklore for some time and were formally stated by Toint (1988). See also Zhu (1992). The first-order criticality measure (12.1.17) was introduced by Conn et al. (1993). Despite its lack of continuity, the norm of the projected gradient on the tangent cone was used to good effect by, for instance, Burke and Moré (1988), Calamai and Moré (1987), Burke, Moré, and Toraldo (1990), and Burke (1990).

## 12.2 An Algorithm for Problems with Convex Constraints

Having considered the geometry of simple feasible regions, we return to the definition of a trust-region algorithm whose purpose is to solve nonlinear optimization problems over such domains. The ideas behind such an algorithm are directly derived from Algorithm BTR, with the main additional requirement being that we shall insist on keeping our iterates feasible, that is,  $x_k \in \mathcal{C}$  for all  $k$ . We shall also need to revise the notion of a “sufficient model decrease”, as the decrease implied by AA.1 may no longer

be achievable because the Cauchy arc may not intersect the feasible region. However, we may now rely on the first-order criticality measure  $\chi(x)$  discussed in the previous section. Thus we will use AA.1b where

$$\pi_k = \chi_k = \chi(x_k) = \left| \min_{\substack{x+d \in \mathcal{C} \\ \|d\| \leq 1}} \langle g_k, d \rangle \right|,$$

instead of AA.1. Observe that, in this definition of the criticality measure, we are already implicitly assuming AM.3. In fact, for the rest of this chapter, we will use all the assumptions of Chapter 6 (restricted to  $\mathcal{C}$  when appropriate), with the exception of AN.1. This latter assumption will not be needed, because we shall assume for simplicity that, for all  $k$ ,

$$\|\cdot\|_k = \|\cdot\|.$$

The trust region is therefore a sphere in the  $\ell_2$  norm, and we obtain that

$$\nu_k^C = \nu_k^S = \nu_k^E = 1$$

for all  $k \geq 0$ . Our algorithm can then be described as Algorithm 12.2.1.

**Algorithm 12.2.1: An algorithm for convex constraints**

**Step 0: Initialization.** An initial point  $x_0 \in \mathcal{C}$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1, \eta_2, \gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Compute  $f(x_0)$  and set  $k = 0$ .

**Step 1: Model definition.** Define a model  $m_k$  in  $\mathcal{C} \cap \mathcal{B}_k$ .

**Step 2: Step calculation.** Compute a step  $s_k$  that sufficiently reduces the model  $m_k$  in the sense of AA.1b and such that  $x_k + s_k \in \mathcal{C} \cap \mathcal{B}_k$ .

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2], \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

### 12.2.1 The Generalized Cauchy Point

Algorithm 12.2.1 remains, at best, of theoretical interest, unless we show that the model decrease implied by AA.1b is achievable when we require our iterates to stay feasible. To do this, as we did in the unconstrained case, we first consider the model decrease that can be achieved by moving along the “steepest-descent” direction, or, in other words, by minimizing just the linear part of the model,  $m_k(x_k) + \langle g_k, s \rangle$ , while at the same time ensuring that  $x_k + s \in \mathcal{C}$ . The locus of the minimizers of this linear part within the trust region is no longer the Cauchy arc, but the projected-gradient path, as we saw in Theorem 12.1.4. We could consider the special case where  $m_k$  is a quadratic model, as we did in Chapter 6, but unfortunately, this is not much help if the constraints are nonlinear, because minimizing a quadratic function subject to nonlinear constraints is far from easy. Instead, we may return to the idea of a “linesearch” along the projected-gradient path<sup>186</sup> to compute our generalized Cauchy point. However, the linesearch we shall consider is slightly different from the Armijo-type technique we used in Section 6.3.3 for the unconstrained case. We shall now consider a Goldstein-type linesearch. Our aim is to determine a  $t_j > 0$  such that, if we set

$$s_k(t) \stackrel{\text{def}}{=} p(t, x_k) - x_k,$$

then both of the conditions

$$\left. \begin{array}{l} \|s_k(t_j)\| \leq \Delta_k \\ \text{and} \\ m_k(p(t_j, x_k)) \leq m_k(x_k) + \kappa_{\text{ubs}} \langle g_k, s_k(t_j) \rangle, \end{array} \right\} \quad (12.2.1)$$

and at least one of

$$\left. \begin{array}{l} \|s_k(t_j)\| \geq \kappa_{\text{frd}} \Delta_k \\ \text{or} \\ m_k(p(t_j, x_k)) \geq m_k(x_k) + \kappa_{\text{lbs}} \langle g_k, s_k(t_j) \rangle \\ \text{or} \\ \|P_{\mathcal{T}(p(t_j, x_k))}[-g_k]\| \leq \kappa_{\text{epp}} \frac{|\langle g_k, s_k(t_j) \rangle|}{\Delta_k} \end{array} \right\} \quad (12.2.2)$$

are satisfied, where<sup>187</sup>

$$0 < \kappa_{\text{ubs}} < \kappa_{\text{lbs}} < 1, \quad \kappa_{\text{frd}} \in (0, 1), \quad \text{and} \quad \kappa_{\text{epp}} \in (0, \frac{1}{2}). \quad (12.2.3)$$

The point

$$x_k^{\text{GC}} = p(t_j, x_k) \quad (12.2.4)$$

is the (approximate)<sup>188</sup> *generalized Cauchy point*. Figure 12.2.1 illustrates two possible

<sup>186</sup>It would be more appropriate to call it a piecewise search in this case.

<sup>187</sup>“lbs” and “ubs” stand for “lower bound on the slope” and “upper bound on the slope”, respectively, “frd” for “fraction of delta”, and “epp” for “end of projected gradient path”.

<sup>188</sup>The word *approximate* is included to indicate that it is obtained by an inexact linesearch procedure, instead of by exact minimization of the model along the projected-gradient path. However, we will not worry about this distinction in what follows, as the notion of exact generalized Cauchy point is only of practical interest when the constraints are very special, such as linear or defined by lower and/or upper bounds on the variables.

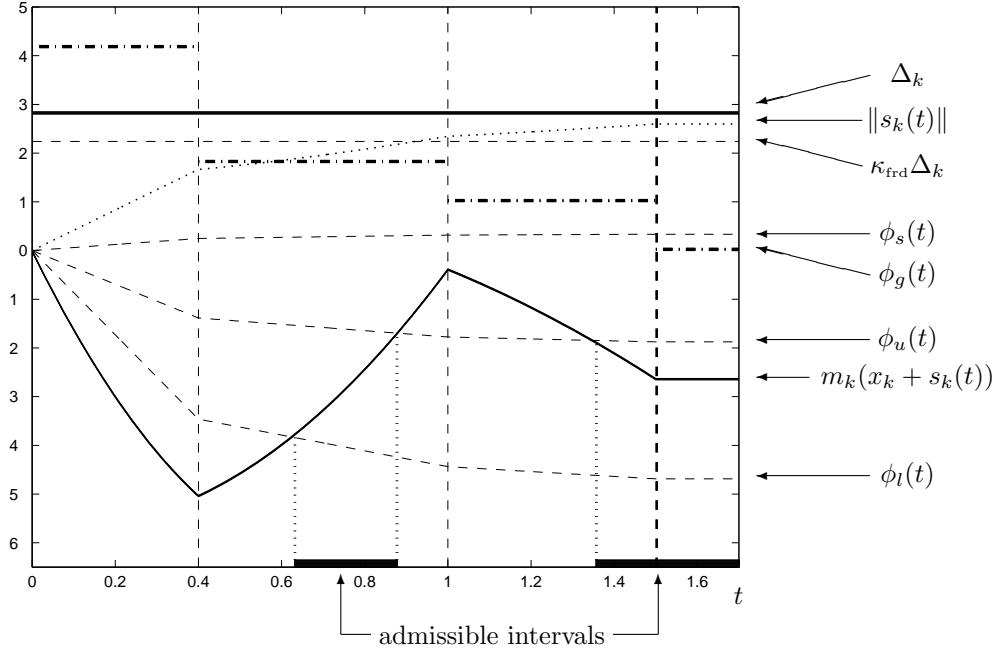


Figure 12.2.1: Admissible intervals for the generalized Cauchy point.

realizations of (12.2.1)–(12.2.2) for a quadratic model. In this figure, we consider the problem of minimizing

$$m_k(x_k + s) = -3.57s_1 - 1.5s_2 - s_3 + s_1s_2 + 3s_2^2 + s_2s_3 - 2s_3^2$$

starting from  $x_k = 0$  and subject to the componentwise bound constraint  $x_k + s \leq 1.5$  and the trust-region bound  $\|s\| \leq 2.8$ . The thick solid piecewise quadratic curve shows the values of  $m_k(p(t, x_k)) = m_k(x_k + s_k(t))$  as a function of the parameter  $t$ . The four thin dashed curves represent (from top to bottom)  $\kappa_{\text{frd}} \Delta_k$ ,

$$\phi_s(t) \stackrel{\text{def}}{=} \kappa_{\text{epp}} \frac{|\langle g_k, s_k(t) \rangle|}{\Delta_k},$$

$$\phi_u(t) \stackrel{\text{def}}{=} m_k(x_k) + \kappa_{\text{ubs}} \langle g_k, s_k(t) \rangle,$$

and

$$\phi_l(t) \stackrel{\text{def}}{=} m_k(x_k) + \kappa_{\text{lbs}} \langle g_k, s_k(t) \rangle.$$

The thick horizontal line in the upper part of the figure gives the value of  $\Delta_k$  and the dotted curve represents  $\|s_k(t)\|$ . The thick dash-pointed staircase curve represents

$$\phi_g(t) \stackrel{\text{def}}{=} \|P_{\mathcal{T}(x_k + s_k(t))}[-g_k]\|.$$

Breakpoints in the piecewise linear projected-gradient path are indicated by three vertical dashed lines, the rightmost and thicker one corresponding to the value of  $t$  beyond which  $s_k(t)$  remains constant. As indicated by thick lines at the bottom of the figure,

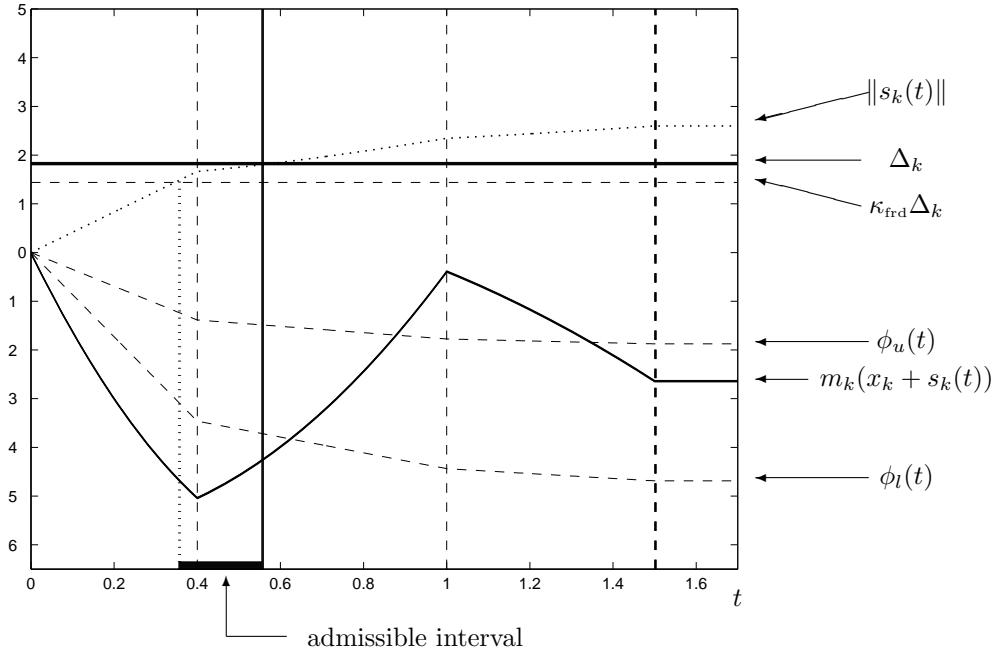


Figure 12.2.2: A third case for the generalized Cauchy point.

two intervals in  $t$  satisfy the conditions (12.2.1)–(12.2.2). In the first, we have that (12.2.1) and the second part of (12.2.2) hold. The second interval in fact consists of two parts: in  $[1.38, 1.5]$ , (12.2.1) and the second part of (12.2.2) again hold, while (12.2.1) and the second and third parts of (12.2.2) hold for  $t \geq 1.5$ . A third case is possible, which is illustrated by Figure 12.2.2, where we consider the same problem as for Figure 12.2.1, but restrict  $\Delta_k$  to 1.8. We have removed the representation of  $\phi_g(t)$  and  $\phi_s(t)$  as they are irrelevant here, but have added a thick vertical line to indicate the maximal  $t$  such that  $s_k(t)$  still belongs to the trust region. In this case, (12.2.1) and the first part of (12.2.2) hold in the admissible interval.

How do we compute a point in the admissible interval(s) in practice? We may simply proceed by bisection on  $t_j$ , as shown in Algorithm 12.2.2.

**Algorithm 12.2.2: Search for the generalized Cauchy point**

**Step 0: Initialization.** The trust-region radius  $\Delta_k$ , the iterate  $x_k \in \mathcal{C}$ , and the model  $m_k(x_k + s)$  are given. The constants  $\kappa_{\text{epp}}$ ,  $\kappa_{\text{lbs}}$ ,  $\kappa_{\text{ubs}}$ , and  $\kappa_{\text{frd}}$  are also given and satisfy the conditions (12.2.3). Set  $t_{\min} = 0$ ,  $t_{\max} = \infty$ ,  $t_0 = \Delta_k / \|g_k\|$ , and  $j = 0$ .

**Step 1: Compute a point on the projected-gradient path.** Set  $p(t_j, x_k) = P_C[x_k - t_j g_k]$  and  $s_k(t_j) = p(t_j, x_k) - x_k$  and evaluate  $m_k(p(t_j, x_k))$ .

**Step 2: Check for the stopping conditions.** If one of the conditions (12.2.1) is violated, then set  $t_{\max} = t_j$  and go to Step 3. Otherwise, if all conditions (12.2.2) are violated, set  $t_{\min} = t_j$  and go to Step 3. Otherwise, set  $x_k^{\text{GC}} = p(t_j, x_k)$  and stop.

**Step 3: Find a new value of the parameter.** If  $t_{\max} = \infty$ , set  $t_{j+1} = 2t_j$ . Otherwise, set  $t_{j+1} = \frac{1}{2}(t_{\min} + t_{\max})$ . Increment  $j$  by 1 and go to Step 1.

Observe that the initial choice of  $t_0$  is arbitrary, so long as  $t_0 > 0$ . Our definition corresponds to the maximum  $t_j$  in the case where the projected-gradient path reduces to the steepest-descent direction within the trust region. For notational convenience, we also define

$$s_k^{\text{GC}} = x_k^{\text{GC}} - x_k.$$

We first verify that the approximate generalized Cauchy point is well defined.

**Theorem 12.2.1** Suppose that AF.1, AC.1, AC.2, AC.7, AO.1, and AM.1 hold. Then Algorithm 12.2.2 terminates in a finite number of iterations.

**Proof.** Suppose first that  $t_{\max} = \infty$  for all  $j$ . This means that (12.2.1) always holds, and thus that  $\|s_k(t_j)\| \leq \Delta_k$  for all  $j$ . Hence (12.1.5) is satisfied, and thus, applying Theorem 12.1.3, (12.1.6) also holds. This in turn implies that the third part of (12.2.2) must hold for  $j$  sufficiently large, since the right-hand side of this condition is strictly positive for  $j = 0$  and, by Theorems 12.1.4 and 12.1.5 (i), non-decreasing. Hence Algorithm 12.2.2 must terminate in a finite number of iterations.

Suppose now that  $t_{\max}$  is reset to a finite value. The existence of an interval of  $t_j$  satisfying the second parts of (12.2.1) and (12.2.2) immediately follows from the continuity of the model  $m_k(x_k + s)$ , guaranteed by AM.1, and the inequalities (12.2.3). But all points in this interval may violate the first part of (12.2.1). However, in this case, we have that the second part of (12.2.2) is violated for  $t$  arbitrarily small and thus for all  $t$  such that  $\|p(t, x_k) - x_k\| \leq \Delta_k$ . Thus, because of (12.2.3), the second part of (12.2.1) also holds for all such  $t$ . Moreover, the interval of  $t$  such that  $\kappa_{\text{frd}}\Delta_k \leq \|p(t, x_k) - x_k\| \leq \Delta_k$  is nonempty by continuity of  $\|p(t, x_k) - x_k\|$ . As a consequence, we have that (12.2.1) hold together with the first part of (12.2.2) for all  $t$  in this interval. That Algorithm 12.2.2 is finite then results from the fact that the desired interval is always contained in  $[t_{\min}, t_{\max}]$  and that the length of this latter interval converges to zero when  $j$  tends to infinity.  $\square$

We now analyse the model decrease obtained at the approximate generalized Cauchy point.

**Theorem 12.2.2** Suppose that AM.1–AM.3 hold and that  $x_k^{\text{GC}}$  is an approximate generalized Cauchy point satisfying (12.2.1), (12.2.2), and (12.2.4). Then

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{dep}} \chi_k \min \left[ \frac{\chi_k}{\beta_k}, \Delta_k, 1 \right], \quad (12.2.5)$$

where  $\chi_k \stackrel{\text{def}}{=} \chi(x_k) = \chi(x_k, 1)$ . Moreover,

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{ubs}} |\langle g_k, s_k^{\text{GC}} \rangle|. \quad (12.2.6)$$

**Proof.** First note that (12.2.6) holds, because of the second part of (12.2.1) and (12.2.4). Then Theorem 12.1.4 gives that

$$|\langle g_k, s_k^{\text{GC}} \rangle| = \chi(x_k, \|s_k^{\text{GC}}\|), \quad (12.2.7)$$

which, together with (12.2.6), yields that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{ubs}} \chi(x_k, \|s_k^{\text{GC}}\|). \quad (12.2.8)$$

We next consider the case where  $\|s_k^{\text{GC}}\| \geq 1$ . In this case, (12.2.8) and Theorem 12.1.5 (i) then ensure that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{ubs}} \chi(x_k, 1) = \kappa_{\text{ubs}} \chi_k. \quad (12.2.9)$$

Suppose from now on that  $\|s_k^{\text{GC}}\| < 1$ . Then (12.2.8) and Theorem 12.1.5 (ii) imply that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{ubs}} \chi_k \|s_k^{\text{GC}}\|. \quad (12.2.10)$$

Consider the case where  $\|s_k^{\text{GC}}\| \geq \kappa_{\text{frd}} \Delta_k$ . Then (12.2.10) gives that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{ubs}} \kappa_{\text{frd}} \chi_k \Delta_k. \quad (12.2.11)$$

On the other hand, if the second part of (12.2.2) holds, then we obtain from Taylor's theorem that

$$m_k(x_k^{\text{GC}}) = m_k(x_k) + \langle g_k, s_k^{\text{GC}} \rangle + \frac{1}{2} \langle s_k^{\text{GC}}, \nabla_{xx} m_k(\xi_k) s_k^{\text{GC}} \rangle$$

for some  $\xi_k$  in the segment  $[x_k, x_k^{\text{GC}}]$ , and therefore that

$$\langle s_k^{\text{GC}}, \nabla_{xx} m_k(\xi_k) s_k^{\text{GC}} \rangle = 2 \left[ m_k(x_k^{\text{GC}}) - m_k(x_k) + |\langle g_k, s_k^{\text{GC}} \rangle| \right] \geq 0.$$

As a consequence, we obtain, from the definition of  $\beta_k$  (p. 124) and from (12.2.2) and (12.2.7), that

$$\beta_k \geq \frac{\langle s_k^{\text{GC}}, \nabla_{xx} m_k(\xi_k) s_k^{\text{GC}} \rangle}{\|s_k^{\text{GC}}\|^2} \geq \frac{2(1 - \kappa_{\text{lbs}}) |\langle g_k, s_k^{\text{GC}} \rangle|}{\|s_k^{\text{GC}}\|^2} \geq \frac{2(1 - \kappa_{\text{lbs}}) \chi_k}{\|s_k^{\text{GC}}\|}, \quad (12.2.12)$$

where we also used the fact that Theorem 12.1.5 (ii) and  $\|s_k^{\text{GC}}\| < 1$  imply the bound

$$\chi(x_k, \|s_k^{\text{GC}}\|) \geq \chi_k \|s_k^{\text{GC}}\|$$

to deduce the last inequality. Now (12.2.12) and (12.2.10) together give that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq 2\kappa_{\text{ubs}}(1 - \kappa_{\text{lbs}})\frac{\chi_k^2}{\beta_k}. \quad (12.2.13)$$

Finally, if the third part of (12.2.2) holds but  $\|s_k^{\text{GC}}\| < \kappa_{\text{frd}}\Delta_k$ , we have, using Theorem 12.1.5 (iii) and the third part of (12.2.2), that

$$\chi(x_k, \min[\kappa_{\text{frd}}\Delta_k, 1]) \leq |\langle g_k, s_k^{\text{GC}} \rangle| + 2\min[\kappa_{\text{frd}}\Delta_k, 1]\|P_{\mathcal{T}(x_k^{\text{GC}})}[-g_k]\| \leq 2|\langle g_k, s_k^{\text{GC}} \rangle|$$

and therefore, using Theorem 12.1.5 (ii), that

$$|\langle g_k, s_k^{\text{GC}} \rangle| \geq \frac{1}{2}\chi(x_k, \min[\kappa_{\text{frd}}\Delta_k, 1]) \geq \frac{1}{2}\kappa_{\text{frd}}\chi_k \min[\Delta_k, 1],$$

and we obtain, using (12.2.6), that

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \frac{1}{2}\kappa_{\text{ubs}}\kappa_{\text{frd}}\chi_k \min[\Delta_k, 1]. \quad (12.2.14)$$

We then obtain (12.2.5) by combining (12.2.9), (12.2.11), (12.2.13), and (12.2.14), and setting  $\kappa_{\text{dep}} \stackrel{\text{def}}{=} \min[\frac{1}{2}\kappa_{\text{ubs}}\kappa_{\text{frd}}, 2\kappa_{\text{ubs}}(1 - \kappa_{\text{lbs}})] < 1$ .  $\square$

### 12.2.2 Convergence to First-Order Critical Points

We now modify slightly our first-order criticality measure to emphasize the similarity between the form of the model decrease at the generalized Cauchy point and AA.1b. More precisely, we define

$$\pi_k \stackrel{\text{def}}{=} \min[1, \chi_k] \leq \chi_k \quad (12.2.15)$$

and notice that we may then rewrite the model decrease at the generalized Cauchy point as

$$m_k(x_k) - m_k(x_k^{\text{GC}}) \geq \kappa_{\text{dep}}\pi_k \min\left[\frac{\pi_k}{\beta_k}, \Delta_k, 1\right] \geq \kappa_{\text{dep}}\pi_k \min\left[\frac{\pi_k}{\beta_k}, \Delta_k\right],$$

where we used the fact that  $\beta_k \geq 1$ , and thus  $\pi_k/\beta_k \leq 1$ , to derive the second inequality. Clearly,  $\pi_k$  is another first-order criticality measure in the sense defined in Section 8.1.1. Thus, if we require the step  $s_k$  to produce a model decrease at least as large as (a fraction of) that obtained at the generalized Cauchy point, as we required in the unconstrained case, we see that this notion of sufficient decrease and (12.2.5)<sup>189</sup> ensure that AA.1b (p. 251) holds for the criticality measure  $\pi_k$ ; that is,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}}\pi_k \min\left[\frac{\pi_k}{\beta_k}, \Delta_k\right].$$

---

<sup>189</sup>We will use (12.2.6) later.

This brings us back to our convergence theory of Chapter 6 again. Applying the conclusions of Section 8.1.2, we thus conclude that

$$\lim_{k \rightarrow \infty} \pi_k = 0,$$

which also implies that

$$\lim_{k \rightarrow \infty} \chi_k = 0.$$

Hence *every limit point of the sequence of iterates generated by Algorithm 12.2.1 is first-order critical*, provided assumptions AF.1, AF.2,<sup>190</sup> AF.3, AC.1, AC.2, AC.7, AO.1, AM.1–AM.4, and AA.1b hold.

## Notes and References for Section 12.2

If the model is quadratic and, additionally, the constraints linear, it is possible to compute the equivalent of the exact generalized Cauchy point by decomposing the projected-gradient path into a sequence of segments, along each of which the model can be expressed as a quadratic function of a single variable. One then considers each of these segments in turn and determines if they contain a local minimizer of the piecewise quadratic resulting from the succession of these one-dimensional quadratics. The search stops as soon as the first local minimum is found. (Note that there may be more than one.) This procedure is especially easy to implement when the constraints are lower and upper bounds on the variables, as we shall see in Section 17.3. This is, for instance, the case in the LANCELOT package of Conn, Gould, and Toint (1992b), which was analysed in Conn, Gould, and Toint (1988a) and described in detail in Conn, Gould, and Toint (1988b). Other software packages also incorporate similar ideas; consider, for example, VE08 and VE10, two routines for optimization subject to bounds that use partitioned quasi-Newton updating (see Section 8.4.1.2), by Toint (1983b, 1987b); BOX-QUACAN, an algorithm by Friedlander, Martínez, and Santos (1994a) (see also Diniz-Ehrhardt, Gomes-Ruggiero, and Santos, 1997), which uses an “easy” subproblem to calculate the Cauchy point; and the TRON package by Lin and Moré (1999a). We also mention the trust-region algorithm of Sartenaer (1995) and Tong and Wang (1999), which take advantage of the structure of the linear constraints when they happen to represent flow conservation equations in a network, and the proposal by Zhu (1999), where nonmonotone techniques in the spirit of Section 10.1 are combined with a method very similar to Algorithm 12.2.1.

We have presented a Goldstein-type linesearch procedure in Section 12.2.1, but a backtracking (Armijo) linesearch of the type used in Section 6.3.3 is also possible. However, we have to extend this technique to allow “forward stepping” (in a manner similar to that already used in Section 10.3.1) as well as backtracking, because we do not know a starting value (from which backtracking can then take place) for the parameter  $t$  of the projected-gradient path. Indeed, there may not be a value of  $t$  for which  $\|p(t, x_k) - x_k\| = \Delta_k$ , and even if such a value exists, we may not know it a priori. This variant was studied by Sartenaer (1993), but we leave the details to the inquisitive reader.

The use of a linesearch along the projected-gradient path has a long history, including contributions by Goldstein (1964), Levitin and Polyak (1966), McCormick (1969), Polyak

---

<sup>190</sup>Observe that we actually only require that  $f(x) \geq \kappa_{\text{lb}} f$  for all  $x \in \mathcal{C}$ , since Algorithm 12.2.1 only evaluates the objective at feasible points.

(1969), Bertsekas (1976, 1982a), Dunn (1980), and Calamai and Moré (1987). This idea is implemented in several software packages, of which L-BFGS-B, a package to minimize nonlinear functions subject to bound constraints, by Byrd et al. (1995) and Zhu et al. (1997), is the best known. It was first introduced in the trust-region framework by Conn, Gould, and Toint (1988a) (for the exact generalized Cauchy point) and Toint (1988) (for the approximate version). The same idea also appeared in several later papers, such as those by Moré (1988) and Moré and Toraldo (1991).

Note that our assumption that  $\|\cdot\|_k = \|\cdot\|$  for all  $k$  is only made to simplify the exposition. All results derived in this chapter may also be derived if we allow general norms for the definition of the trust region, provided we then require AN.1 to hold, that is, if we make sure that these norms are uniformly equivalent. This more general derivation may be found in Conn, Gould, Sartenaer, and Toint (1996b).

We also note that a “regularized” version of the algorithm described above can be proved to be convergent to first-order critical points when the assumption that the objective function is twice-continuously differentiable is replaced by the requirement that its gradient be Lipschitz continuous (see Liang and Xu, 1997, for details). Finally, we note that a trust-region method for linearly constrained problems using simplicial decomposition was discussed by Feng (1998, 1999).

## 12.3 Active Constraints Identification

### 12.3.1 Further Assumptions

We now consider the evolution of the set of constraints that are active at the successive iterates generated by Algorithm 12.2.1,  $\mathcal{A}(x_k)$ . More precisely, we will show that this set “settles down” in the sense that it remains constant (and equal to the set of constraints active at a solution of the problem) after a finite number of iterations. However, this result depends on further assumptions on both the problem and the algorithm. We start by making our constraint qualification condition more specific by requiring linear independence of the active constraints normals. Let  $\mathcal{L}_*$  be the set of all limit points of the sequence  $\{x_k\}$ .

**AO.1c** For all  $x_* \in \mathcal{L}_*$ , the Jacobian of active constraint gradients,  $\nabla_x c_{\mathcal{A}(x_*)}(x_*)$ , is of full rank.

As indicated in Section 3.2.2, this assumption subsumes AO.1 and AO.2. In particular, AC.1, AC.7, and AO.1c together ensure that Theorem 3.2.10 (p. 46) holds, and the normal cone is thus spanned by the outward normals to each of the active constraints.

We complete our assumptions on the problem by assuming a nondegeneracy condition at every limit point  $x_* \in \mathcal{L}_*$ .

**AO.4b** For all  $x_* \in \mathcal{L}_*$ , one has that  $-\nabla_x f(x_*) \in \text{ri}\{\mathcal{N}(x_*)\}$ .

This condition may be viewed as a generalization of the strict complementarity condition. In particular, it implies, together with AO.1c, that for every  $x_* \in \mathcal{L}_*$ , there exists

a unique set of strictly positive Lagrange multipliers  $\{y_{i*}\}_{i \in \mathcal{A}(x_*)}$  such that

$$\nabla_x f(x_*) = \sum_{i \in \mathcal{A}(x_*)} y_{i*} \nabla_x c_i(x_*). \quad (12.3.1)$$

As we will show that eventually the optimal active set is identified by the generalized Cauchy point, we need to make sure that this information is not lost when computing the step  $s_k$ . This is guaranteed if we require all constraints that are active at the generalized Cauchy point  $x_k^{GC}$  to remain so at the trial point  $x_k + s_k$ .

**AA.5** All constraints active at the generalized Cauchy point  $x_k^{GC}$  remain active at  $x_k + s_k$ ; that is,

$$\mathcal{A}(x_k^{GC}) \subseteq \mathcal{A}(x_k + s_k).$$

Observe that constraints inactive at  $x_k^{GC}$  may become active at  $x_k + s_k$ .

### Notes and References for Subsection 12.3.1

The nondegeneracy condition AO.4b was introduced by Dunn (1987), and a discussion of its relations with strict complementarity may be found in Burke and Moré (1988). We refer the interested reader to Conn et al. (1993) or to Burke and Moré (1988), Burke, Moré, and Toraldo (1990), Burke (1990), Friedlander, Martínez, and Santos (1994a), and Lescrenier (1991) for further discussion.

### 12.3.2 The Geometry of the Set of Limit Points

Using the assumptions just stated, we now investigate the geometry of the set of limit points  $\mathcal{L}_*$ . We first show the remarkable fact (which depends critically upon assumption AO.4b) that each connected subset of  $\mathcal{L}_*$  only spreads onto a single face of the feasible region.

**Lemma 12.3.1** Suppose that AF.1, AC.1, AC.2, AC.7, AO.1c, and AO.4b hold. Then, for each connected component of limit points  $\mathcal{L}(x_*) \subseteq \mathcal{L}_*$ , there exists a set  $\mathcal{A}_* \subseteq \{1, \dots, m\}$  for which

$$\mathcal{A}(x_*) = \mathcal{A}_*$$

for all  $x_* \in \mathcal{L}(x_*)$ .

**Proof.** Consider two limit points  $x_*$  and  $v_*$  in  $\mathcal{L}(x_*)$  such that

$$\mathcal{A}(x_*) \neq \mathcal{A}(v_*) \quad (12.3.2)$$

and assume, without loss of generality, that there exists a  $j \in \{1, \dots, m\}$  such that  $j \in \mathcal{A}(v_*)$  but  $j \notin \mathcal{A}(x_*)$ . Because of the path-connectivity of  $\mathcal{L}(x_*)$ , we know that there exists a continuous path  $z(t)$  such that

$$z(0) = x_*, \quad z(1) = v_*, \quad z(t) \in \mathcal{L}(x_*) \text{ for all } t \in [0, 1].$$

Using the continuity of  $z(\cdot)$  and  $c_j(\cdot)$ , the condition (12.3.2) and the definition of  $j$  also ensure the existence of  $t_+ \in (0, 1]$  such that

$$j \in \mathcal{A}(z(t_+)) \text{ and } j \notin \mathcal{A}(z(t)) \text{ for all } t \in [0, t_+). \quad (12.3.3)$$

Let us consider a sequence  $\{t_j\}$  in the interval  $[0, t_+)$  converging to  $t_+$  and such that  $\mathcal{A}(z(t_j))$  is constant and equal to  $\mathcal{A}_-$ , say, for all  $j$ . Equation (12.3.1) implies that

$$\nabla_x f(z(t_j)) = \sum_{i \in \mathcal{A}_-} y_i^-(t_j) \nabla_x c_i(z(t_j)) \quad (12.3.4)$$

for all  $t_j$  and for some uniquely defined multipliers  $y_i^-(t_j) > 0$ . We now wish to show by contradiction that the sequences  $\{y_i^-(t_j)\}$  are bounded for all  $i \in \mathcal{A}_-$ . So suppose otherwise, that the sequence of vectors  $\{y^-(t_j)\}$  is unbounded, where these vectors have  $\{y_i^-(t_j)\}_{i \in \mathcal{A}_-}$  for fixed  $j$  as components. In this case, we can select a subsequence  $\{t_\ell\} \subseteq \{t_j\}$  such that

$$\|y^-(t_\ell)\| \rightarrow \infty \text{ and } \frac{y^-(t_\ell)}{\|y^-(t_\ell)\|} \rightarrow \hat{y},$$

where  $\hat{y}$  is normalized and has at least one strictly positive component. We then obtain from (12.3.4) that

$$\frac{\nabla_x f(z(t_\ell))}{\|y^-(t_\ell)\|} = \sum_{i \in \mathcal{A}_-} \frac{y_i^-(t_\ell)}{\|y^-(t_\ell)\|} \nabla_x c_i(z(t_\ell)),$$

which gives in the limit that

$$0 = \sum_{i \in \mathcal{A}_-} \hat{y}_i \nabla_x c_i(z(t_+)), \quad (12.3.5)$$

where we have used the continuity of  $z(\cdot)$ , AF.1, AC.1, and AC.7. If we now define

$$\mathcal{A}_+ \stackrel{\text{def}}{=} \mathcal{A}(z(t_+)),$$

we note that (12.3.3) and the fact that the set  $\{x \in \mathbb{R}^n \mid \mathcal{A}_- \subseteq \mathcal{A}(x)\}$  is closed ensure that  $\mathcal{A}_- \subset \mathcal{A}_+$ . Therefore, because of AO.1c and the fact that  $z(t_+) \in \mathcal{L}_*$ , we may deduce from (12.3.5) that all the components of  $\hat{y}$  are zero, which we just indicated to be impossible. Hence the sequence  $\{y^-(t_j)\}$  must be bounded, as must be the sequences of its components. From each of these components' subsequences, we may thus extract converging subsequences with limit points  $y_i^-$ . Using the continuity of  $z(\cdot)$ , AF.1, AC.1, and AC.7 and again taking the limit in (12.3.4) for these subsequences, we obtain that

$$\nabla_x f(z(t_+)) = \sum_{i \in \mathcal{A}_-} y_i^- \nabla_x c_i(z(t_+)). \quad (12.3.6)$$

On the other hand, (12.3.1) implies that

$$\nabla_x f(z(t_+)) = \sum_{i \in \mathcal{A}_+} y_i^+ \nabla_x c_i(z(t_+)) \quad (12.3.7)$$

for some uniquely defined set of multipliers  $y_i^+ > 0$ . But the fact that  $\mathcal{A}_- \subset \mathcal{A}_+$  ensures that (12.3.6) and (12.3.7) cannot hold together. Our initial assumption (12.3.2) is thus impossible, which proves the lemma.  $\square$

Thus we may associate a specific active set with each connected component of limit points. If  $\mathcal{L} \subseteq \mathcal{L}_*$  is such a connected component, we shall denote its associated active set by  $\mathcal{A}(\mathcal{L})$  in what follows.

After showing that different active sets cannot appear in a single connected component of limit points, we now show that connected components of limit points are well separated when all limit points are finite.

**Lemma 12.3.2** Suppose that AF.1, AC.1, AC.2, AC.7, AO.1c, AO.4b, and AI.1 hold. Then there exists a  $\psi \in (0, 1)$  such that

$$\text{dist}(x_*, \mathcal{L}') \geq \psi,$$

for every  $x_* \in \mathcal{L}_*$  and each compact connected component of limit points  $\mathcal{L}'$  such that  $\mathcal{A}(\mathcal{L}') \neq \mathcal{A}(x_*)$ .

**Proof.** Consider any  $x_* \in \mathcal{L}_*$ . With this  $x_*$ , we can associate the sets

$$\mathcal{D}_i \stackrel{\text{def}}{=} \{x \in \Omega \mid i \in \mathcal{A}(x)\},$$

for  $i \notin \mathcal{A}(x_*)$ , where  $\Omega$  is the compact set in which the iterates are assumed to remain by AI.1. For each  $x_* \in \mathcal{L}_*$ , there is only a finite number of such sets, and each of them is compact, because they are all closed by definition and contained in  $\Omega$ . Because of Lemma 12.3.1, the sets  $\mathcal{D}_i$  and  $\mathcal{L}(x_*)$  are disjoint for all  $i \notin \mathcal{A}(x_*)$ . From the compactness of  $\mathcal{L}_*$ , we then deduce the existence of a  $\psi \in (0, 1)$  such that

$$\min_{x_* \in \mathcal{L}_*} \min_{i \notin \mathcal{A}(x_*)} \min_{x \in \mathcal{D}_i} \|x_* - x\| \geq \psi, \quad (12.3.8)$$

which yields the desired result since  $\mathcal{L}' \subset \mathcal{L}_* \subseteq \Omega$ .  $\square$

Unfortunately, the assumption that the sequence of iterates remains bounded (AI.1) cannot be removed, as is shown by the following example. Consider the simple two-dimensional problem

$$\min_{x_1, x_2 \geq 0} f(x_1, x_2) = \sin^2 x_1 + x_2(x_2 - e^{-x_1})^2.$$

It can be easily verified that  $f(x_1, x_2) \geq 0$  for all feasible points and that  $f(x_1^*, x_2^*) = 0$  if and only if  $(x_1^*, x_2^*)$  has the form

$$(j\pi, 0) \text{ or } (j\pi, e^{-j\pi})$$

for some nonnegative integer  $j$ . Hence these points are all second-order critical points of the problem having the same objective function value and satisfying AO.1c and AO.4b.

They can be limit points of the sequence of iterates. Furthermore, the constraint  $x_2 \geq 0$  is (strongly) active at all critical points of the form  $(j\pi, 0)$ , while no constraint is active at the critical points of the form  $(j\pi, e^{-j\pi})$ . However, the distance between  $(j\pi, 0)$  and  $(j\pi, e^{-j\pi})$  tends to zero when  $j$  tends to infinity, and Lemma 12.3.2 cannot hold. Of course, if we enforce our assumption, say, by imposing a finite upper bound  $\kappa \geq 0$  on  $x_1$ , then we obtain that the lemma holds for a value of the lower bound  $\psi$  given by  $\psi = e^{-\lfloor \frac{\kappa}{\pi} \rfloor \pi}$ .

We next use this “separation result” to show that, for  $k$  sufficiently large, every iterate  $x_k$  lies in the neighbourhood of a well-defined connected component of limit points. To make this notion precise, we need to formally state what we mean by the neighbourhood (vicinity) of a set  $\mathcal{E}$  of radius  $\delta$ . In what follows, we use the intuitive definition

$$\mathcal{V}(\mathcal{E}, \delta) = \{x \in \mathbb{R}^n \mid \text{dist}(x, \mathcal{E}) \leq \delta\}.$$

We may also, at this point, show that all constraints that are not binding for this component are also inactive at  $x_k$ . These two results are formally stated in the following useful theorem.

**Theorem 12.3.3** Suppose that AF.1, AC.2, AC.7, AO.1c, AO.4b, and AI.1 hold. Then there exist constants  $\delta \in (0, \frac{1}{4}\psi)$ ,  $\psi \in (0, 1)$ , and an index  $k_1 \geq 0$  such that, for  $k \geq k_1$ , there is a compact connected component of limit points  $\mathcal{L}_{*k} \subseteq \mathcal{L}_*$  such that

$$x_k \in \mathcal{V}(\mathcal{L}_{*k}, \delta) \quad (12.3.9)$$

and

$$\mathcal{A}(x) \subseteq \mathcal{A}(\mathcal{L}_{*k}) \text{ for all } x \in \mathcal{V}(\mathcal{L}_{*k}, \delta). \quad (12.3.10)$$

**Proof.** Because of AI.1, we may divide the complete sequence  $\{x_k\}$  into a number of subsequences, each of which converges to a given connected component of limit points. For  $k$  sufficiently large,  $x_k$  therefore lies in the vicinity of one such connected component, say  $\mathcal{L}_{*k}$ . The inclusion (12.3.9) then follows for  $\delta$  small enough and for  $k$  sufficiently large. We then obtain (12.3.10) by using (12.3.8) and imposing the additional requirement that  $\delta < \frac{1}{4}\psi$ .  $\square$

The next step in our development is to prove that, if an iterate  $x_k$  is sufficiently close to its associated connected component of limit points, but  $x_k^{\text{GC}}$  has an incomplete<sup>191</sup> set of active constraints, then  $x_k$  is bounded away from criticality in the sense that  $\pi_k$  is bounded away from zero by a small constant independent of  $k$ .

<sup>191</sup>With respect to that associated with the connected component of limit points, as specified by Lemma 12.3.1.

**Theorem 12.3.4** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.3, and AI.1 hold. Then there exists  $k_2 \geq k_1$  (where  $k_1$  is as defined in Theorem 12.3.3 with  $\delta < \frac{1}{2}$ ) such that, if there is a  $j \in \{1, \dots, m\}$  with

$$j \in \mathcal{A}(\mathcal{L}_{*k}) \text{ and } j \notin \mathcal{A}(x_k^{\text{GC}}) \quad (12.3.11)$$

for some  $k \geq k_2$ , then

$$\pi_k \geq \epsilon_* \quad (12.3.12)$$

for some  $\epsilon_* \in (0, 1)$  independent of  $k$  and  $j$ .

**Proof.** Consider, for a given  $x_* \in \mathcal{L}_*$  with  $\mathcal{A}(x_*) \neq \emptyset$  and a given  $i \in \mathcal{A}(x_*)$ , the quantity

$$\pi_{*i}(x_*) \stackrel{\text{def}}{=} \left| \min_{\substack{x_* + d \in \mathcal{C}_{\{i\}} \\ \|d\| \leq \frac{1}{2}}} \langle \nabla_x f(x_*), d \rangle \right|,$$

where  $\mathcal{C}_{\{i\}}$  is defined by

$$\mathcal{C}_{\{i\}} \stackrel{\text{def}}{=} \bigcap_{j \in \{1, \dots, m\} \setminus \{i\}} \mathcal{C}_j.$$

The value of  $\pi_{*i}(x_*)$  is the magnitude of the decrease obtained by minimizing the linearized objective function from  $x_*$  in a sphere of radius  $\frac{1}{2}$  when dropping the  $i$ th active constraint. Because of AC.7, AO.1c, and AO.4b, one has that  $\pi_{*i}(x_*) > 0$  for all choices of  $x_* \in \mathcal{L}_*$  and  $i \in \mathcal{A}(x_*)$ . Theorem 3.2.8 (p. 44) and the continuity of  $\nabla_x f(\cdot)$  (ensured by AF.1) also guarantee that  $\pi_{*i}(\cdot)$  is continuous. We first minimize  $\pi_{*i}(x_*)$  on the compact set of all  $x_* \in \mathcal{L}_*$  such that  $i \in \mathcal{A}(x_*)$ . For each such set, this produces a strictly positive result. We next take the smallest of these results over all  $i$  such that  $i \in \mathcal{A}(x_*)$  for some  $x_* \in \mathcal{L}_*$ , yielding a strictly positive lower bound  $2\epsilon_*$ . In short,

$$\min_i \min_{x_*} \pi_{*i}(x_*) \geq 2\epsilon_* \quad (12.3.13)$$

for some  $\epsilon_* \in (0, 1)$  independent of  $k$ ,  $j$ , and  $\delta$ .

Now reduce  $\delta$ , if necessary, to ensure that

$$\kappa_{\text{ufh}} \delta \leq \epsilon_*, \quad (12.3.14)$$

and consider  $k \geq k_1$ . Then, by Theorem 12.3.3, we can associate with  $x_k$  a unique connected component of limit points  $\mathcal{L}_{*k}$  such that (12.3.9) holds. We then select a particular  $x_{*k} \in \mathcal{L}_{*k} \cap \mathcal{V}(x_k, \delta)$ , which ensures that

$$\{x_{*k} + d \in \mathcal{C}_{\{i\}} \mid \|d\| \leq \frac{1}{2}\} \subset \{x_k + d \in \mathcal{C}_{\{i\}} \mid \|d\| \leq 1\} \quad (12.3.15)$$

for all  $i \in \{1, \dots, m\}$ , where we used the bound  $\delta < \frac{1}{2}$ . This inclusion is illustrated in Figure 12.3.1 (where we have not assumed that  $x_* \in \mathcal{L}_{*k}$ ).

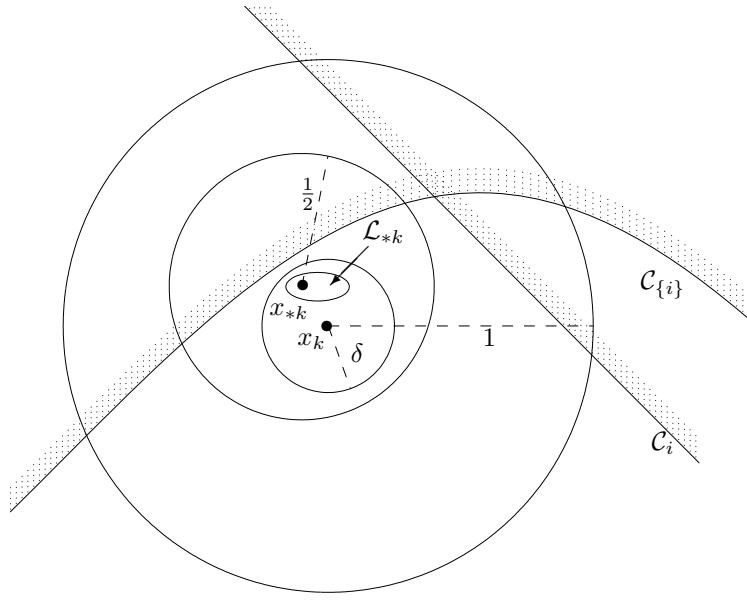


Figure 12.3.1: The inclusion (12.3.15).

Given a  $k \geq k_1$  and such that  $x_k$  satisfies (12.3.11), we now distinguish two cases. The first is when  $\pi_k \geq \pi_{*j}(x_{*k})$ , in which case (12.3.12) immediately follows from (12.3.13). The second is when  $\pi_k < \pi_{*j}(x_{*k})$ . If  $\chi_k \geq 1$ , then  $\pi_k = 1$  by definition, and (12.3.12) again follows since  $\epsilon_* \in (0, 1)$ . Suppose therefore that  $\chi_k < 1$ , in which case  $\pi_k = \chi_k$ , and define  $d_k$  and  $d_{*k}$  as two vectors satisfying

$$\pi_k = -\langle g_k, d_k \rangle, \quad \|d_k\| \leq 1, \quad x_k + d_k \in \mathcal{C}$$

and

$$\pi_{*j}(x_{*k}) = -\langle \nabla_x f(x_{*k}), d_{*k} \rangle, \quad \|d_{*k}\| \leq \frac{1}{2}, \quad x_{*k} + d_{*k} \in \mathcal{C}_{\{j\}}.$$

We can write, using the Cauchy–Schwarz inequality, that

$$\begin{aligned} 0 < \pi_{*j}(x_{*k}) - \pi_k &= \langle g_k, d_k \rangle - \langle \nabla_x f(x_{*k}), d_{*k} \rangle \\ &= \langle g_k, d_k - d_{*k} \rangle + \langle g_k - \nabla_x f(x_{*k}), d_{*k} \rangle \quad (12.3.16) \\ &\leq \langle g_k, d_k - d_{*k} \rangle + \frac{1}{2} \|g_k - \nabla_x f(x_{*k})\|. \end{aligned}$$

Now combining (12.3.15) and the definitions of  $\pi_k$ ,  $d_k$ , and  $d_{*k}$ , we obtain that

$$\langle g_k, d_k \rangle = -\pi_k \leq \langle g_k, d_{*k} \rangle.$$

Substituting this last inequality in (12.3.16), we choose  $k_2 \geq k_1$  sufficiently large to ensure that, for  $k \geq k_2$ ,

$$0 < \pi_{*j}(x_{*k}) - \pi_k \leq \|g_k - \nabla_x f(x_{*k})\| \leq \kappa_{\text{ufh}} \|x_k - x_{*k}\| \leq \kappa_{\text{ufh}} \delta \leq \epsilon_*,$$

where we used AM.3, AF.3, the definition of  $x_{*k}$ , and (12.3.14). The inequality (12.3.12) then follows from (12.3.13).  $\square$

### 12.3.3 The Identification of Active Constraints

With the results on the geometry of limit points proved so far, we are now in position to prove that, given a limit point  $x_*$  of the sequence of iterates generated by Algorithm 12.2.1, the set of constraints that are active at  $x_*$  is identified by the algorithm in a finite number of iterations. This is the object of this section.

We first show that, if the trust-region radius is small and the correct active set has not been identified at the generalized Cauchy point  $x_k^{\text{GC}}$  for  $k$  sufficiently large (which implies, by Theorem 12.3.4, that (12.3.12) holds), then the  $k$ th iteration is successful.

**Lemma 12.3.5** Suppose that AF.1, AF.3, AM.1–AM.4, and AA.1b hold. Suppose furthermore that

$$\Delta_k \leq \frac{\kappa_{\text{mdc}} \pi_k (1 - \eta_2)}{\kappa_{\text{ubh}}}$$

for some  $k \geq k_2$ . Then iteration  $k$  is very successful and  $\Delta_{k+1} \geq \Delta_k$ .

**Proof.** The proof of this result is identical to that of Theorem 6.4.2 (p. 134), where  $\pi_k$  plays the role of  $\|g_k\|$  and AA.1b that of AA.1.  $\square$

We also need the result that the gradient projected onto the tangent cone at a point  $u$  having the correct active set tends to zero as both this point and the iterates approach a connected component of limit points.

**Lemma 12.3.6** Suppose that AF.1–AF.3, AM.1–AM.4, AC.1, AC.2, AC.7, AO.4b, and AI.1 hold. Let  $\mathcal{K}$  be the index set of an infinite subsequence such that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \text{dist}(x_k, \mathcal{L}) = \lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \|u_k - x_k\| = 0$$

for some connected component of limit points  $\mathcal{L} \subseteq \mathcal{L}_*$  and some sequence  $\{u_k\}_{k \in \mathcal{K}}$  such that  $u_k \in \mathcal{C}$  and  $\mathcal{A}(u_k) = \mathcal{A}(\mathcal{L})$  for all  $k \in \mathcal{K}$ . Then one has that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} P_{\mathcal{T}(u_k)}[-g_k] = 0.$$

**Proof.** We first note that Theorem 3.2.8 (p. 44) and the continuity of the constraint normals (ensured by AC.1) imply the continuity of the operators  $P_{\mathcal{T}(\cdot)}$  and  $P_{\mathcal{N}(\cdot)}$  as functions of  $\{u \mid \mathcal{A}(u) = \mathcal{A}(\mathcal{L})\}$  in a sufficiently small neighbourhood of  $\mathcal{L}$ . We also observe that the Moreau decomposition (p. 36) of  $-g_k$  gives that

$$-g_k = P_{\mathcal{T}(u_k)}[-g_k] + P_{\mathcal{N}(u_k)}[-g_k].$$

The assumptions of the lemma and AO.4b then give the desired limit by continuity.  $\square$

Among the finitely many active sets  $\{\mathcal{A}(x_*)\}_{x_* \in \mathcal{L}_*}$ , we now consider a maximal one and denote it by  $\mathcal{A}_*$ . This is to say that  $\mathcal{A}_* = \mathcal{A}(x_*)$  for some  $x_* \in \mathcal{L}_*$ , and that

$$\mathcal{A}_* \not\subset \mathcal{A}(u_*) \quad (12.3.17)$$

for any  $u_* \in \mathcal{L}_*$ . We have now set the scene for a crucial technical lemma which states that the correct active set is identified at least on a subsequence of successful iterations.

**Lemma 12.3.7** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.4, AA.1b, AA.5, and AI.1 hold. Then there exists a subsequence  $\{k_i\}$  of successful iterations such that, for  $i$  sufficiently large,

$$\mathcal{A}(x_{k_i}) = \mathcal{A}_*. \quad (12.3.18)$$

**Proof.** We define the subsequence  $\{k_j\}$  as the sequence of successful iterations whose iterates approach limit points with their active set equal to  $\mathcal{A}_*$ ; that is,

$$\{k_j\} \stackrel{\text{def}}{=} \{k \in \mathcal{S} \mid \mathcal{A}(\mathcal{L}_{*k}) = \mathcal{A}_*\},$$

and assume, for the purpose of obtaining a contradiction, that

$$\mathcal{A}(x_{k_j+1}) \neq \mathcal{A}_* \quad (12.3.19)$$

for  $j$  large enough. Assume now, again for the purpose of contradiction, that

$$\mathcal{A}_* \subseteq \mathcal{A}(x_{k_j}^{\text{GC}}) \quad (12.3.20)$$

for such a  $j$ . Using successively AA.5, (12.3.19), and Theorem 12.3.3, we obtain that, for  $j$  sufficiently large,

$$\mathcal{A}_* \subset \mathcal{A}(\mathcal{L}_{*k_j+1}),$$

which is impossible because of (12.3.17). Hence (12.3.20) cannot hold, and there must be a  $p_j \in \mathcal{A}_* = \mathcal{A}(\mathcal{L}_{*k_j})$  such that  $p_j \notin \mathcal{A}(x_{k_j}^{\text{GC}})$  for  $j$  large enough. From Theorem 12.3.4, we then deduce that (12.3.12) holds for  $j$  sufficiently large. But the fact that iteration  $k_j$  is successful, together with AA.1b and AM.4, implies that

$$f(x_{k_j}) - f(x_{k_j+1}) \geq \eta_1 \kappa_{\text{mdc}} \epsilon_* \min \left[ \frac{\epsilon_*}{\beta_{k_j}}, \Delta_{k_j} \right] \geq \eta_1 \kappa_{\text{mdc}} \epsilon_* \min \left[ \frac{\epsilon_*}{\kappa_{\text{ubh}}}, \Delta_{k_j} \right]$$

for  $j$  large enough, and thus that

$$\lim_{j \rightarrow \infty} \Delta_{k_j} = 0. \quad (12.3.21)$$

We therefore obtain that

$$\|s_{k_j}\| \leq \Delta_{k_j} \leq \frac{1}{2}\delta < \frac{1}{4}\psi$$

for  $j$  larger than  $j_1$ , say. But this last inequality and Theorems 12.3.3 and 12.3.4 imply that  $x_{k_j+1}$  cannot jump to the vicinity of any other connected component of limit points with a different active set, and hence  $x_{k_j+1}$  belongs to  $\mathcal{V}(\mathcal{L}, \delta)$  again for some  $\mathcal{L}$  such that  $\mathcal{A}(\mathcal{L}) = \mathcal{A}_*$ . The same property also holds for the next successful iteration, say  $x_{k_j+q}$ , and we have that  $\mathcal{A}(\mathcal{L}_{*k_j+q}) = \mathcal{A}_*$ . Therefore, the subsequence  $\{k_j\}$  is identical to the complete sequence of successful iterations with  $k \geq k_{j_1}$ . Hence we may deduce from (12.3.21) that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{S}}} \Delta_k = 0. \quad (12.3.22)$$

In particular, we have that

$$\Delta_k \leq \frac{\gamma_1^2 \kappa_{\text{mdc}} \epsilon_* (1 - \eta_2)}{2 \kappa_{\text{ubh}}} \quad (12.3.23)$$

for  $k \in \mathcal{S}$  sufficiently large. But the mechanism of the algorithm and (12.3.22) also give the limit

$$\lim_{k \rightarrow \infty} \Delta_k = 0. \quad (12.3.24)$$

As a consequence, we note that, for  $k$  large enough,  $x_k$ ,  $x_k^{\text{GC}}$ , and  $x_k + s_k$  all belong to  $\mathcal{V}(\mathcal{L}, \delta)$  for a single connected component of limit points  $\mathcal{L}$ .

We also note that Lemma 12.3.5, the fact that (12.3.12) now holds for  $k \in \mathcal{S}$ , and (12.3.22) together give that

$$k \in \mathcal{S} \implies \Delta_{k+1} \geq \Delta_k, \quad (12.3.25)$$

for  $k$  large enough. We can therefore deduce the desired contradiction from (12.3.24) and (12.3.25) if we can prove that all iterations are eventually successful.

Suppose therefore that this is not the case. It is then possible to find a subsequence  $\mathcal{K}$  of sufficiently large  $k$  such that

$$k \notin \mathcal{S} \text{ and } k+1 \in \mathcal{S}. \quad (12.3.26)$$

Note that, because of the mechanism of Step 4 of Algorithm 12.2.1, one has that

$$\Delta_k \leq \frac{\Delta_{k+1}}{\gamma_1} \leq \frac{\gamma_1 \kappa_{\text{mdc}} \epsilon_* (1 - \eta_2)}{2 \kappa_{\text{ubh}}}, \quad (12.3.27)$$

where we use (12.3.23) to deduce the last inequality. Now, if one has that

$$\mathcal{A}(x_k^{\text{GC}}) \subset \mathcal{A}(\mathcal{L}) = \mathcal{A}_*, \quad (12.3.28)$$

then Theorem 12.3.3 and Lemma 12.3.5 together with (12.3.27) imply that  $k \in \mathcal{S}$ , which contradicts (12.3.26). Hence (12.3.28) cannot hold, and AA.5 with Theorem 12.3.4 give that

$$\mathcal{A}(x_k + s_k) = \mathcal{A}(x_k^{\text{GC}}) = \mathcal{A}(\mathcal{L}) \quad (12.3.29)$$

for  $k$  sufficiently large. Observe now that, since  $k \notin \mathcal{S}$ , one has that  $x_{k+1} = x_k$ , and hence that

$$\begin{aligned} m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) &= m_{k+1}(x_k + s_{k+1}) - m_k(x_k + s_k) \\ &= \langle -g_k, s_k - s_{k+1} \rangle + \frac{1}{2} \langle s_{k+1}, \nabla_{xx} m_{k+1}(\xi_{k+1}) s_{k+1} \rangle \\ &\quad - \frac{1}{2} \langle s_k, \nabla_{xx} m_k(\xi_k) s_k \rangle \\ &\geq \langle -g_k, s_k - s_{k+1} \rangle - \frac{1}{2} \beta_{k+1} \Delta_{k+1}^2 - \frac{1}{2} \beta_k \Delta_k^2 \\ &\geq \langle -g_k, s_k - s_{k+1} \rangle - \frac{1}{2} \kappa_{\text{ubh}} \left(1 + \frac{1}{\gamma_1^2}\right) \Delta_{k+1}^2, \end{aligned}$$

where we have successively used  $\xi_k \in [x_k, x_k + s_k]$ ,  $\xi_{k+1} \in [x_{k+1}, x_{k+1} + s_{k+1}]$ , the Cauchy–Schwarz inequality, AM.4, the fact that the steps are bounded by the trust-region radius, and the mechanism of Step 4 of Algorithm 12.2.1. But

$$\begin{aligned} \langle -g_k, s_k - s_{k+1} \rangle &= \langle P_{\mathcal{T}(x_k+s_k)}[-g_k], s_k - s_{k+1} \rangle + \langle P_{\mathcal{N}(x_k+s_k)}[-g_k], s_k - s_{k+1} \rangle \\ &\geq -\|P_{\mathcal{T}(x_k+s_k)}[-g_k]\| \|s_k - s_{k+1}\| \\ &\quad - \langle P_{\mathcal{N}(x_k+s_k)}[-g_k], P_{\mathcal{T}(x_k+s_k)}[s_{k+1} - s_k] \rangle \\ &\geq -\|P_{\mathcal{T}(x_k+s_k)}[-g_k]\| \left( \|s_k\| + \|s_{k+1}\| \right) \\ &\geq -\left(1 + \frac{1}{\gamma_1^2}\right) \Delta_{k+1} \|P_{\mathcal{T}(x_k+s_k)}[-g_k]\| \end{aligned}$$

for all  $k \in \mathcal{K}$ , where we have used the Moreau decomposition (p. 36) of  $-g_k$ , the fact that  $s_{k+1} - s_k \in \mathcal{T}(x_k + s_k)$ , the polar of  $\mathcal{N}(x_k + s_k)$ , and, as above, the fact that the steps are bounded by the trust-region radius, the fact that  $\gamma_1 < 1$ , and the mechanism of Step 4. Combining the last two chains of inequalities, we obtain that

$$\begin{aligned} m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) &\geq -\left(1 + \frac{1}{\gamma_1^2}\right) \Delta_{k+1} \left[ \|P_{\mathcal{T}(x_k+s_k)}[-g_k]\| + \frac{1}{2} \kappa_{\text{ubh}} \Delta_{k+1} \right] \end{aligned}$$

for such  $k$ . We may now recall (12.3.24) and, because of the equality (12.3.29), apply Lemma 12.3.6 (with  $u_k = x_k + s_k$ ) and therefore deduce from this last inequality that

$$m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) \geq -\frac{1}{2} \kappa_{\text{mdc}} \epsilon_* \Delta_{k+1}$$

for  $k$  large enough in  $\mathcal{K}$ . On the other hand, AA.1b, the fact that  $k+1 \in \mathcal{S}$ , (12.3.24), and the bounds (12.3.12) and  $\beta_k \leq \kappa_{\text{ubh}}$  imply that

$$f(x_{k+1}) - m_{k+1}(x_{k+1} + s_{k+1}) \geq \kappa_{\text{mdc}} \epsilon_* \Delta_{k+1}.$$

Hence, we obtain that

$$\begin{aligned} f(x_k) - m_k(x_k + s_k) &= f(x_{k+1}) - m_{k+1}(x_{k+1} + s_{k+1}) \\ &\quad + m_{k+1}(x_{k+1} + s_{k+1}) - m_k(x_k + s_k) \\ &\geq \frac{1}{2} \kappa_{\text{mdc}} \epsilon_* \Delta_{k+1} \\ &\geq \frac{1}{2} \kappa_{\text{mdc}} \gamma_1 \epsilon_* \Delta_k \end{aligned}$$

for all  $k \in \mathcal{K}$  sufficiently large. But then, using the definition of  $\rho_k$ , Theorem 6.4.1 (p. 133) (with  $\nu_k^s = 1$ ), and (12.3.27), we obtain that

$$|\rho_k - 1| \leq \frac{2\kappa_{\text{ubh}}}{\kappa_{\text{mdc}}\gamma_1\epsilon_*} \Delta_k \leq 1 - \eta_2,$$

and hence that  $\rho_k \geq \eta_2$  for all  $k \in \mathcal{K}$  large enough, which contradicts (12.3.26). The condition (12.3.26) is thus impossible for  $k$  sufficiently large. All iterations are eventually very successful, which produces the desired contradiction.

As a consequence, (12.3.19) cannot hold for all  $j$ , and we obtain that there exists a subsequence  $\{k_p\} \subseteq \{k_j\}$  such that, for all  $p$ ,

$$\mathcal{A}_* = \mathcal{A}(x_{k_p+1}) = \mathcal{A}(x_{k_p+q(k_p)}),$$

where  $k_p + q(k_p)$  is the first successful iteration after iteration  $k_p$ . The lemma is thus proved if we choose  $\{k_i\} = \{k_p + q(k_p)\}$ .  $\square$

The last step in our analysis of the active set identification is to show that, once found, the maximal active set  $\mathcal{A}_*$  cannot be abandoned for sufficiently large  $k$ . This is the essence of the final theorem in this section.

**Theorem 12.3.8** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.4, AA.1b, AA.5, and AI.1 hold. Then one has that

$$\mathcal{A}(x_*) = \mathcal{A}_* \quad (12.3.30)$$

for all  $x_* \in \mathcal{L}_*$ , and

$$\mathcal{A}(x_k) = \mathcal{A}(x_k^{\text{GC}}) = \mathcal{A}_* \quad (12.3.31)$$

for all  $k$  sufficiently large.

**Proof.** Consider  $\{k_i\}$ , the subsequence of successful iterates such that (12.3.18) holds, as given by Lemma 12.3.7. Suppose furthermore that this subsequence is restricted to sufficiently large indices, that is,  $k_i \geq k_2$  for all  $i$ . Suppose finally that there exists a subsequence of  $\{k_i\}$ , say  $\{k_p\}$ , such that, for each  $p$ , there is a  $j_p$  such that

$$j_p \in \mathcal{A}(x_{k_p}) \text{ and } j_p \notin \mathcal{A}(x_{k_p+1}).$$

Now Theorem 12.3.3 and (12.3.18) give that  $\mathcal{A}(\mathcal{L}_{*k_p}) = \mathcal{A}_*$ . Using this observation and AA.5, we obtain that

$$j_p \in \mathcal{A}(\mathcal{L}_{*k_p}) \text{ and } j_p \notin \mathcal{A}(x_{k_p}^{\text{GC}})$$

for all  $p$ . But Theorem 12.3.4 then ensures that

$$\pi_k \geq \epsilon_* \quad (12.3.32)$$

for all  $p$ . Combining this inequality with AA.1b and AM.4, we obtain that, for all  $p$ ,

$$f(x_{k_p}) - f(x_{k_p+1}) \geq \eta_1 \kappa_{\text{mdc}} \epsilon_* \min \left[ \frac{\epsilon_*}{\kappa_{\text{ubh}}}, \Delta_k \right],$$

which implies that

$$\lim_{k \rightarrow \infty} \Delta_k = 0. \quad (12.3.33)$$

Applying AA.1b and AM.4 and using (12.3.32), we obtain that

$$f(x_{k_p}) - m_{k_p}(x_{k_p} + s_{k_p}) \geq \kappa_{\text{mdc}} \epsilon_* \Delta_{k_p} \quad (12.3.34)$$

for all  $p$  sufficiently large. On the other hand, we have that, for all  $k$ ,

$$\begin{aligned} f(x_k) - m_k(x_k + s_k) &\leq |\langle g_k, s_k \rangle| + \kappa_{\text{ubh}} \|s_k\|^2 \\ &\leq \chi(x_k, \|s_k\|) + \kappa_{\text{ubh}} \|s_k\|^2 \\ &\leq \frac{\chi(x_k, \|s_k\|)}{\|s_k\|} \Delta_k + \kappa_{\text{ubh}} \|s_k\|^2, \end{aligned} \quad (12.3.35)$$

where we used the definition of  $\chi(\cdot, \cdot)$ . Combining (12.3.34), (12.3.35) taken at  $k = k_p$ , applying Theorem 12.1.5 (iv), and dividing both sides by  $\Delta_k$ , we obtain that

$$\kappa_{\text{mdc}} \epsilon_* \leq \|P_{\mathcal{T}(x_{k_p})}[-g_k]\| + \kappa_{\text{ubh}} \Delta_{k_p}. \quad (12.3.36)$$

Assuming that the sequence  $\{x_{k_p}\}$  converges to some  $x_*$  in some  $\mathcal{L}$  (or taking a further subsequence if necessary), using (12.3.33) and Lemma 12.3.6 (with  $\mathcal{K} = \{k_p\}$ ,  $u_k = x_k$  and  $\mathcal{A}(\mathcal{L}) = \mathcal{A}_*$ ), we deduce that (12.3.36) is impossible for  $p$  large enough. As a consequence, no such subsequence  $\{k_p\}$  exists, and we have that, for large  $i$ ,

$$\mathcal{A}_* \subseteq \mathcal{A}(x_{k_i}^{\text{GC}}) \subseteq \mathcal{A}(x_{k_i+1}) \subseteq \mathcal{A}(\mathcal{L}_{*k_i+1}),$$

where the last inclusion follows from Theorem 12.3.3. But the maximality of  $\mathcal{A}_*$  then imposes that

$$\mathcal{A}_* = \mathcal{A}(x_{k_i}^{\text{GC}}) = \mathcal{A}(x_{k_i+1}) = \mathcal{A}(\mathcal{L}_{*k_i+1})$$

for  $i$  sufficiently large. Hence we obtain that, for  $i$  large enough,

$$\mathcal{A}(x_{k_i+q}) = \mathcal{A}_*,$$

where  $k_i + q$  is the index of the first successful iteration after iteration  $k_i$ . Hence  $k_i + q \in \{k_i\}$ . We can therefore repeatedly apply this reasoning and deduce that

$$\{k_i\} = \{k \in \mathcal{S} \mid k \text{ is sufficiently large}\}$$

and also that  $\mathcal{A}(x_k) = \mathcal{A}_*$  for all  $k \in \mathcal{S}$  large enough, hence proving (12.3.31). Moreover,  $\mathcal{A}_*$  is then the only possible active set for the limit points, which proves (12.3.30).  $\square$

We now know that all iterates  $x_k$  remain in the optimal face

$$\mathcal{F}_* = \{x \in \mathbb{R}^n \mid \mathcal{A}(x) = \mathcal{A}_*\}$$

for  $k$  sufficiently large. This then implies that we may reformulate the convergence to first-order critical points in terms of the projected gradient.

**Corollary 12.3.9** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.4, AA.1b, AA.5, and AI.1 hold. Then

$$\lim_{k \rightarrow \infty} \|P_{\mathcal{T}(x_k)}[-g_k]\| = 0.$$

**Proof.** This immediately results from Lemma 12.3.6 (with  $\mathcal{K} = \{1, 2, \dots\}$  and  $u_k = x_k$ ) and Theorem 12.3.8.  $\square$

Note that the conclusion of this corollary is stronger than the limits  $\lim_{k \rightarrow \infty} \pi_k = 0$  or  $\lim_{k \rightarrow \infty} \chi_k = 0$ , as can be seen from (12.2.15) and Theorem 12.1.5 (iv).

## Notes and References for Section 12.3

The identification of the active constraints has been studied in the context of projected-gradient algorithms for optimization problems involving simple bound or linear constraints by several authors (see, for instance, Calamai and Moré, 1987, and Burke and Moré, 1988). The first results in the context of trust-region methods appeared in Conn, Gould, and Toint (1988a) for the case where the only constraints are bounds on the variables. An application to the linear least-squares case may be found in Bierlaire, Toint, and Tuyttens (1991). Our exposition, covering general convex constraints, is based on Sartenaer (1991) and Conn et al. (1993), which also covers the case where the norm defining the trust region is allowed to vary from iteration to iteration (subject to AN.1) and the use of approximate derivatives in the sense of Section 8.4. A similar theory is developed in Sartenaer (1993) and in Conn, Gould, Sartenaer, and Toint (1996b) for a variant of Algorithm 10.2.2 (p. 368) that handles convex constraints. The identification properties of trust-region algorithms for infinite-dimensional problems (see Section 8.3) are discussed by Toint (1988) and Ulbrich and Ulbrich (1997) for the case of pointwise bounds.

It is interesting to note that, although our discussion has derived the result that the projected gradient converges to zero (Corollary 12.3.9) from the fact that the optimal active set is identified (Theorem 12.3.8), it is also possible to establish the implication in the other direction. This is the elegant approach followed by Burke and Moré (1988), Burke (1990), Burke, Moré, and Toraldo (1990), and Burke and Moré (1994). The last of these papers points out that identification results may also be derived without the nondegeneracy assumption AO.4b—in this case, only strongly active constraints are identified—an observation first made by Lescrenier (1991). The papers of Chen (1995) and Facchinei, Júdice, and Soares (1998)

also make significant use of constraint identification techniques, the latter in conjunction with nonmonotone methods.

## 12.4 Convergence to Second-Order Critical Points

Once the correct active set has been identified, one may view the optimization as “unconstrained” in the optimal face corresponding to this active set. If the constraints are linear, this face is an affine subspace, and one may expect the sequence  $\{x_k\}$  to converge to second-order critical points under conditions similar to those of Chapter 6, although restricted to the optimal face. This is especially important since Theorem 6.5.5 (p. 146) then allows for a local analysis of the rate of convergence of the algorithm. However, if the constraints are nonlinear, the question is technically more difficult, mostly because directions of search (for instance, in the determination of the eigenpoint) are not contained in the optimal face, but rather in its tangent plane. It is the purpose of this section to examine how the second-order convergence results of Chapter 6, Sections 6.5 and 6.6, can be adapted to this more complex situation.

### 12.4.1 The Role of the Lagrangian

For general smooth convex constraints, as specified by AC.1, AC.2, and AC.7, the second-order necessary conditions involve the vector  $y$  of Lagrange multipliers (see Theorem 3.2.5 [p. 41]). Thus checking second-order criticality will require estimates of these multipliers, since in this case we are interested in the eigenstructure of the Hessian of the Lagrangian

$$\nabla_{xx}\ell(x_*, y_*) = \nabla_{xx}f(x_*) - \sum_{i=1}^m [y_*]_i \nabla_{xx}c_i(x_*),$$

where  $x_*$  is a limit point and  $y_*$  the associated vector of multipliers. We know from the discussion of Section 12.2.2 that all limit points of the sequence of iterates generated by Algorithm 12.2.1 are first-order critical, and AC.6, AO.1c, and AO.4b then guarantee, as we already noted above, the existence and uniqueness of the vector  $y_*$ , given  $x_*$ . However,  $y_*$  is typically not known until  $x_*$  has been determined, which means that instead we will have to estimate multipliers  $y_k$  from properties associated with the iterate  $x_k$ . In what follows, we use the *least-squares multipliers*  $y^{\text{LS}}(x)$ , which are uniquely defined, for  $x \in \mathcal{C}$ , as

$$y^{\text{LS}}(x) = \arg \min_y \|\nabla_x f(x) - A(x)^T y\|,$$

where  $A(x)$  is the Jacobian of the constraints at  $x$ , and where the minimization is carried out with the additional consideration that  $y$  is a *consistent* vector of multipliers; that is,

$$y \geq 0 \text{ and } i \notin \mathcal{A}(x) \Rightarrow [y]_i = 0.$$

The uniqueness of  $y^{\text{LS}}(x)$  of  $x$  in the neighbourhood of a given limit point  $x_*$  results from AC.7 and AO.1c. In other words, using Theorem 3.2.10 (p. 46), we have that

$$-A(x_k)^T y_k^{\text{LS}} = P_{\mathcal{N}(x_k)}[-g_k],$$

where we have defined  $y_k^{\text{LS}} = y^{\text{LS}}(x_k)$ , and therefore, using AM.3 and the Moreau decomposition (p. 36) of  $-g_k$  at  $x_k$ ,

$$\begin{aligned} -\nabla_x \ell(x_k, y_k^{\text{LS}}) &= -g_k + \sum_{i=1}^m [y_k^{\text{LS}}]_i \nabla_x c_i(x_k) \\ &= -g_k + A(x_k)^T y_k^{\text{LS}} \\ &= -g_k - P_{\mathcal{N}(x_k)}[-g_k] \\ &= P_{\mathcal{T}(x_k)}[-g_k]. \end{aligned} \tag{12.4.1}$$

Moreover, the continuity of the projection operator and AO.1c imply that  $y^{\text{LS}}(x)$  is a continuous function of  $x$ , and thus that, for all sequences of feasible points  $\{u_k\}$ ,

$$\lim_{k \rightarrow \infty} y^{\text{LS}}(u_k) = y(u_*) \quad \text{when} \quad \lim_{k \rightarrow \infty} u_k = u_*.$$

In particular,

$$\lim_{k \rightarrow \infty} y_k^{\text{LS}} = y_*.$$

As expected, we then obtain from Corollary 12.3.9 that

$$\nabla_x \ell(x_*, y_*) = \lim_{k \rightarrow \infty} \nabla_x \ell(x_k, y_k^{\text{LS}}) = 0.$$

The role of the Lagrangian  $\ell(x_k + s, y)$  is also important because it coincides with the objective function on the face defined by  $\mathcal{A}(x_k)$ . Indeed, since  $c_i(x_k + s) = 0$  for  $i \in \mathcal{A}(x_k)$  and  $\mathcal{A}(x_k + s) = \mathcal{A}(x_k)$ , we deduce that

$$f(x_k + s) = f(x_k + s) - \sum_{i \in \mathcal{A}_*} [y]_i c_i(x_k + s) = f(x_k + s) - \sum_{i=1}^m [y]_i c_i(x_k + s) = \ell(x_k + s, y)$$

for any consistent vector of multipliers  $y$ . In particular,

$$f(x_k) = \ell(x_k, y_k^{\text{LS}}) \tag{12.4.2}$$

for all  $k$ . Similarly, under the same conditions,

$$m_k(x_k + s) = m_k(x_k + s) - \sum_{i \in \mathcal{A}_*} [y_k]_i c_i(x_k + s) \stackrel{\text{def}}{=} m_k^\ell(x_k + s). \tag{12.4.3}$$

Hence, provided  $x_k + s_k$  remains<sup>192</sup> in the same face as  $x_k$ , the model may be interpreted either as a model of the objective function, as usual, or as a model of the Lagrangian. We may formalize this observation as follows.

---

<sup>192</sup>Observe that this condition must hold for all successful iterations, because of Theorem 12.3.8.

**Lemma 12.4.1** Suppose that AC.1, AC.7, AO.1c, AO.4b, AM.1, and AM.3 hold. Then, provided  $\mathcal{A}(x_k + s_k) = \mathcal{A}(x_k)$ , we have that

$$m_k(x_k + s_k) = m_k(x_k) - \langle P_{T(x_k)}(-g_k), s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} m_k^\ell(\xi_k) s_k \rangle$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ .

**Proof.** Using successively (12.4.3) for  $s = s_k$ , AM.1 and the mean value theorem, AM.3, (12.4.3) for  $s = 0$ , and (12.4.1), we obtain that

$$\begin{aligned} m_k(x_k + s_k) &= m_k^\ell(x_k + s_k) \\ &= m_k^\ell(x_k) + \langle \nabla_x m_k^\ell(x_k), s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} m_k(\xi_k) s_k \rangle \\ &= m_k(x_k) - \langle P_{T(x_k)}(-g_k), s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} m_k(\xi_k) s_k \rangle \end{aligned}$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ , as desired.  $\square$

It is important to note, at this stage, that we will not need the Lagrange multipliers if the constraints of the problem are linear, that is, if  $\mathcal{C}$  is described by bounds on the variables or is specified as a convex polyhedral set. Indeed,  $\nabla_{xx} c_i(x)$  is identically zero for all linear constraints and thus  $\nabla_{xx} m_k^\ell(\xi_k) = \nabla_{xx} m_k(\xi_k)$ . However, we will not consider the case of linear constraints very specifically in what follows, but simply remember that the calculation of  $y_k^{\text{LS}}$  is then irrelevant.

### 12.4.2 Convex Models

As in Chapter 6, we first show the convergence of the complete sequence of iterates to a single limit point when the models are asymptotically convex and the Hessian of the objective function is nonsingular in the tangent plane.

We start by reformulating Lemma 6.5.1 (p. 140), using the model of the Lagrangian function.

**Lemma 12.4.2** Suppose that AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1, and AA.1b hold. Suppose furthermore that  $\mathcal{A}(x_k) = \mathcal{A}(x_k + s_k)$  and that, for all  $x \in [x_k, x_k + s_k]$ ,

$$\langle s_k, \nabla_{xx} m_k^\ell(x) s_k \rangle \geq \epsilon \|s_k\|^2 \quad (12.4.4)$$

for some  $\epsilon > 0$ . Then

$$\|s_k\| \leq \frac{2 \langle P_{T(x_k)}[-g_k], s_k \rangle}{\epsilon \|s_k\|}. \quad (12.4.5)$$

**Proof.** See the proof of Lemma 6.5.1 (p. 140). Because of Lemma 12.4.1, the

model decrease obtained at  $x_k + s_k$  can be written as

$$m_k(x_k) - m_k(x_k + s_k) = \langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle - \frac{1}{2} \langle s_k, \nabla_{xx} m_k^\ell(\xi_k) s_k \rangle$$

for some  $\xi_k$  in the segment  $[x_k, x_k + s_k]$ . Suppose first that  $m_k(x_k + s_k) = m_k(x_k)$ . Then AA.1b implies that  $\pi_k = 0$  and thus that  $s_k = 0$ . If, on the other hand,  $m_k(x_k + s_k) < m_k(x_k)$ , we have that  $s_k \neq 0$ . We then define

$$\phi(t) \stackrel{\text{def}}{=} m_k(x_k) - m_k(x_k + ts_k) = t \langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle - \frac{1}{2} t^2 \langle s_k, \nabla_{xx} m_k^\ell(\xi_k) s_k \rangle$$

for  $t > 0$ . But (12.4.4) and the fact that  $s_k \neq 0$  ensure that  $\langle s_k, \nabla_{xx} m_k^\ell(\xi_k) s_k \rangle > 0$ , and hence  $\phi$  is a concave quadratic function. Moreover,  $\phi(0) = 0$  and  $\phi(1) > 0$  by construction. As a consequence,

$$t_* = \arg \max_t \phi(t) \geq \frac{1}{2},$$

where we used the symmetry of a quadratic with respect to its maximum to derive the last inequality (see Figure 6.5.1 [p. 140]). But a direct calculation yields that

$$t_* = \frac{\langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle}{\langle s_k, \nabla_{xx} m_k^\ell(\xi_k) s_k \rangle} \leq \frac{\langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle}{\epsilon \|s_k\|^2},$$

where we have used (12.4.4) to derive the last part of this bound. Therefore, we deduce that (12.4.5) holds.  $\square$

Our objective is now to follow the outline of Chapter 6 by investigating the behaviour of the sequence of iterates when the models are asymptotically convex on the plane tangent to the active constraints. For this purpose, we define, for any  $x \in \mathcal{C}$ ,  $N(x)$  to be an  $n \times (n - |\mathcal{A}(x)|)$  matrix whose columns form a basis of the plane tangent at  $x$  to the face containing  $x$ , and suppose that  $N(x)$  is a locally continuous function of  $x$  (see Section 4.4.2).

**Theorem 12.4.3** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.4, AA.1b, AA.5, and AI.1 hold. Suppose furthermore that there is a  $\kappa_{\text{smh}} > 0$  such that

$$\liminf_{k \rightarrow \infty} \min_{x \in \mathcal{B}_k \cap \mathcal{C}} \langle s_k, \nabla_{xx} m_k^\ell(x) s_k \rangle \geq \kappa_{\text{smh}} \|s_k\|^2.$$

Suppose finally that  $N(x_*)^T \nabla_{xx} \ell(x_*, y_*) N(x_*)$  is positive definite. Then the complete sequence of iterates  $\{x_k\}$  converges to a single first-order critical point  $x_*$ .

**Proof.** If there are only finitely many successful iterations, then the desired conclusion immediately follows. Otherwise, consider  $\{x_{k_i}\}$ , a subsequence of iterates converging to  $x_*$ . The criticality of  $x_*$  is ensured by Theorem 6.4.6 and we may

assume, without loss of generality, that the subsequence  $\{k_i\}$  consists only of successful iterations, which yields

$$x_{k_i+1} = x_{k_i} + s_{k_i}$$

for all  $i$ . Furthermore, we then deduce from Theorem 12.3.8 that both  $x_{k_i}$  and  $x_{k_i+1}$  belong to  $\mathcal{F}_*$  for  $i$  sufficiently large, say, for  $i \geq i_1$ . We now choose  $\delta_1 > 0$  small enough to ensure that  $N(x)^T \nabla_{xx} \ell(x, y^{\text{LS}}(x)) N(x)$  is uniformly positive definite in  $\mathcal{V}(x_*, \delta_1) \cap \mathcal{F}_*$ . We then introduce

$$\delta_* \stackrel{\text{def}}{=} \frac{\kappa_{\text{smh}} \delta_1}{4 + \epsilon} < \delta_1$$

and define  $\ell_{\mathcal{X}}$  to be the largest value of the Lagrangian such that

$$\mathcal{X} \stackrel{\text{def}}{=} \{x \in \mathcal{V}(x_*, \delta_1) \cap \mathcal{F}_* \mid \ell(x, y^{\text{LS}}(x)) \leq \ell_{\mathcal{X}}\} \subseteq \mathcal{V}(x_*, \delta_*),$$

which is possible because the positive definiteness of  $N(x)^T \nabla_{xx} \ell(x, y^{\text{LS}}(x)) N(x)$  guarantees the strict convexity of  $\ell(x, y^{\text{LS}}(x))$  as a function of  $x$  in this set.

We then choose  $i_2 \geq i_1$  such that

$$x_{k_i} \in \mathcal{X}$$

for all  $i \geq i_2$ , and also that

$$\min_{x \in \mathcal{B}_{k_i} \cap \mathcal{C}} \langle s_k, \nabla_{xx} m_k^\ell(x) s_k \rangle \geq \frac{1}{2} \kappa_{\text{smh}} \|s_k\|^2$$

and

$$\|P_{\mathcal{T}(x_k)}[-g_k]\| \leq \delta_*$$

for all  $k \geq k_{i_2}$ , the last condition being possible because of Corollary 12.3.9. We may then apply Lemma 12.4.2 with  $\epsilon = \frac{1}{2} \kappa_{\text{smh}}$  and deduce from the Cauchy–Schwarz inequality that

$$\|s_{k_i}\| \leq \frac{4 \langle P_{\mathcal{T}(x_{k_i})}[-g_{k_i}], s_{k_i} \rangle}{\kappa_{\text{smh}} \|s_{k_i}\|} \leq \frac{4}{\kappa_{\text{smh}}} \|P_{\mathcal{T}(x_{k_i})}[-g_{k_i}]\| \leq \frac{4 \delta_*}{\kappa_{\text{smh}}}$$

for  $i \geq i_2$ . This in turns implies, using the triangle inequality, that

$$\|x_{k_i+1} - x_*\| \leq \|x_{k_i} - x_*\| + \|s_{k_i}\| \leq \left(1 + \frac{4}{\kappa_{\text{smh}}}\right) \delta_* = \delta_1.$$

We now recall that  $k_i \in \mathcal{S}$ , and therefore that

$$\ell(x_{k_i+1}, y_{k_i+1}^{\text{LS}}) = f(x_{k_i+1}) < f(x_{k_i}) = \ell(x_{k_i}, y_{k_i}^{\text{LS}}) \leq \ell_{\mathcal{X}},$$

where we used (12.4.2) twice. Hence  $x_{k_i+1} \in \mathcal{X}$  and all conditions that were satisfied at  $x_{k_i}$  are again satisfied at the next successful iteration after  $k_i$ . The argument can therefore be applied recursively to show that

$$x_{k_i+j} \in \mathcal{X} \subseteq \mathcal{V}(x_*, \delta_1)$$

for all  $j \geq 1$ . Since  $\delta_1$  is arbitrarily small, this proves convergence of the complete sequence  $\{x_k\}$  to  $x_*$ .  $\square$

Thus, we find, as in the unconstrained case, that the use of asymptotically convex models may induce convergence to a single limit point. Observe that convexity of the Lagrangian is not required for all directions of  $\mathbb{R}^n$ , but we merely need the curvature of this function to be positive for steps that preserve the current active set. If the constraints are linear, these steps lie in the tangent plane.

Our next step is to analyse under what circumstances the trust-region radius is bounded away from zero. As in Chapter 6, we obtain this result when the model and the objective function asymptotically coincide up to second order (as in AM.5b). Whether it remains true for general convex constraints is still an open question. For the rest of this section, we restrict our attention to the simpler case where the constraints are linear, in which case  $\mathcal{C}$  is a convex polyhedron. We start by deducing a crucial lower bound on the model decrease obtained from our analysis of the properties of the generalized Cauchy point.

**Lemma 12.4.4** Suppose that AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1, and AA.1b hold. Suppose furthermore that  $\mathcal{A}(x_k) = \mathcal{A}(x_k + s_k)$  and that, for all  $x \in [x_k, x_k + s_k]$ ,

$$\langle s_k, \nabla_{xx} m_k^\ell(x) s_k \rangle \geq \epsilon \|s_k\|^2$$

for some  $\epsilon > 0$ . Suppose finally that all constraints  $c_i(x)$  with  $i \in \mathcal{A}(x_k)$  are linear. Then

$$m_k(x_k) - m_k(x_k + s_k) \geq \delta \|s_k\|^2$$

for some  $\delta > 0$ .

**Proof.** From Theorem 12.2.2, equation (12.2.6), and the fact that we require a model reduction at  $x_k + s_k$  which is at least a (fraction of) that obtained at the generalized Cauchy point, we obtain that, for some  $\kappa \in (0, 1)$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa |\langle g_k, s_k \rangle|. \quad (12.4.6)$$

Now the fact that  $\mathcal{A}(x_k + s_k) = \mathcal{A}(x_k)$  and that the active constraints are linear implies that

$$|\langle g_k, s_k \rangle| = \langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle,$$

since  $s_k$  is then orthogonal to  $\mathcal{N}(x_k)$ . We therefore deduce from this equality and (12.4.6) that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa \langle P_{\mathcal{T}(x_k)}[-g_k], s_k \rangle \geq \frac{1}{2} \kappa \epsilon \|s_k\|^2,$$

where we have used Lemma 12.4.2 to derive the last inequality. The desired bound then follows by defining

$$\delta \stackrel{\text{def}}{=} \frac{1}{2} \kappa \epsilon.$$

□

We are now in position to state our final result on the asymptotic success of the iterations.

**Theorem 12.4.5** Suppose that AF.1–AF.3, AC.1, AC.2, AC.7, AO.1c, AO.4b, AM.1–AM.4, AM.5b, AA.1b, and AI.1 hold. Suppose furthermore that  $\mathcal{A}(x_k) = \mathcal{A}(x_k + s_k)$  and that, for all  $x \in [x_k, x_k + s_k]$ ,

$$\langle s_k, \nabla_{xx} \ell(x, y^{\text{LS}}(x)) s_k \rangle \geq \epsilon \|s_k\|^2$$

for some  $\epsilon > 0$  and for all  $k$  sufficiently large. Suppose finally that all constraints  $c_i(x)$  with  $i \in \mathcal{A}(x_k)$  are linear. Then all iterations are eventually very successful and  $\Delta_k$  is bounded away from zero.

**Proof.** We first observe that  $\pi_k$  tends to zero, as we concluded in Section 12.2.2. Therefore, AM.5b and the continuity of  $y_k^{\text{LS}}$  as a function of  $x_k$  (guaranteed by AO.1c) ensure that

$$\lim_{k \rightarrow \infty} \max_{x \in [x_k, x_k + s_k]} \|\nabla_{xx} m_k^\ell(x) - \nabla_{xx} \ell(x, y^{\text{LS}}(x))\| = 0, \quad (12.4.7)$$

and therefore that

$$\langle s_k, \nabla_{xx} m_k^\ell(x) s_k \rangle \geq \frac{1}{2} \epsilon \|s_k\|^2 \quad (12.4.8)$$

for  $k$  sufficiently large and all  $x \in [x_k, x_k + s_k]$ . We now apply Lemma 12.4.2 and deduce, after applying the Cauchy–Schwarz inequality, that

$$\|s_k\| \leq \frac{4}{\epsilon} \|P_{\mathcal{T}(x_k)}[-g_k]\|.$$

But Corollary 12.3.9 then ensures that

$$\lim_{k \rightarrow \infty} \|s_k\| = 0. \quad (12.4.9)$$

Moreover, (12.4.8) and Lemma 12.4.4 then yield that

$$m_k(x_k) - m_k(x_k + s_k) \geq \delta \|s_k\|^2 \quad (12.4.10)$$

for some  $\delta > 0$  and all  $k$  sufficiently large. The conditions (12.4.9) and (12.4.10) then allow us to apply Lemma 6.5.4 (p. 145) (with AM.5 replaced by (12.4.7)), which yields the desired conclusions.  $\square$

The assumptions of this last theorem are (indeed, maybe too) strong, but not unrealistic. If we consider the case where  $m_k$  is quadratic, then our assumption on uniform minimum curvature in the segment  $[x_k, x_k + s_k]$  is automatically satisfied. Moreover, the fact that  $x_k + s_k$  remains in  $\mathcal{F}(x_k)$  is also reasonable, because we may expect, in the case where the sequence of iterates converges to a single limit point, that  $x_k^{\text{GC}} \in \mathcal{F}(x_k) = \mathcal{F}_*$ . This, combined with the requirement that  $\mathcal{A}(x_k + s_k) \subseteq \mathcal{A}(x_k^{\text{GC}})$  (i.e., AA.5), then ensures that this assumption holds.

### 12.4.3 Nonconvex Models

Can we enforce convergence of Algorithm 12.2.1 to second-order critical points, even if the constraints are nonlinear? This is what we examine now. Because, in the unconstrained case, this property depends on the requirement that the model reduction be at least (a fraction of) that obtained at the eigenpoint, we expect that we will have to rely here on similar concepts. However, there is now the additional difficulty, namely, that a direction of negative curvature for the Hessian of the Lagrangian in the plane tangent to the active constraints at  $x_k$  may only intersect the feasible domain at  $x_k$ , because of the possible nonlinearity of the constraint functions. As we have already done for the negative gradient direction, the natural idea is thus to project such a direction onto the feasible region and to define the “generalized eigenpoint” by a suitable search along the associated projected path. For simplicity, we will again consider only the general case where our model  $m_k$  might not be quadratic and therefore only describe a “projected” version of the analysis of the approximate eigenpoint.

We start by determining a direction of negative curvature for the Hessian of the model Lagrangian  $\nabla_{xx}m_k^\ell(x_k)$  along the plane tangent to the feasible set at  $x_k$ . It is important to observe that such a direction cannot, in general, be calculated without obtaining (numerical) estimates of the multipliers,<sup>193</sup>  $y_k$ . This differs from the situation of the previous section, where we used the existence of the least-squares multipliers  $y_k^{\text{LS}}$  to derive our results, but without requiring that we actually calculate them anywhere in the algorithm. Of course, we still have to specify what conditions we shall require of  $y_k$ , which is the object of the next assumption.

**AA.6** The multipliers estimates  $y_k$  are consistent for  $k$  sufficiently large, and

$$\lim_{k \rightarrow \infty} \|y_k - y_k^{\text{LS}}\| = 0 \quad \text{whenever} \quad \lim_{k \rightarrow \infty} \pi_k = 0.$$

The main reason for requiring (asymptotic) consistency for the multipliers estimates is that this choice ensures that the model  $m_k$  and model Lagrangian  $m_k^\ell$  are identical on the optimal face, as we have already seen in (12.4.3). However, asymptotic consistency alone is not sufficient to ensure the same sort of convergence condition on the Hessian of the Lagrangian as we required of the Hessian of the objective function in the unconstrained case (AM.5b), namely, that  $\{\nabla_{xx}m_k(x_k) - \sum_{i=1}^m [y_k]_i \nabla_{xx}c_i(x_k)\}$  converges to  $\nabla_{xx}f(x_*) - \sum_{i=1}^m [y_*]_i \nabla_{xx}c_i(x_*)$  if  $\{x_k\}$  converges to  $x_*$ . Such a condition is guaranteed if we require AM.5b and the last part of AA.6, because the definition of  $y_k^{\text{LS}}$  then ensures that

$$\lim_{k \rightarrow \infty} y_k^{\text{LS}} = y_* \quad \text{if} \quad \lim_{k \rightarrow \infty} x_k = x_*. \quad (12.4.11)$$

The choice  $y_k = y_k^{\text{LS}}$  clearly satisfies AA.6 if the sequence of iterates remains bounded, but this might be relatively costly to compute and we may wish to use less costly approximations.<sup>194</sup>

---

<sup>193</sup>Thus the determination of a suitable negative curvature direction requires both an estimation of the multipliers  $y_k$  and a (possibly implicit) eigenvalue analysis of the resulting reduced Hessian of the Lagrangian  $N(x_k)^T [\nabla_{xx}f(x_k) - \sum_{i=1}^m [y_k]_i \nabla_{xx}c_i(x_k)] N(x_k)$ .

<sup>194</sup>Such as first-order multiplier estimates; see Section 14.2.

Having calculated  $y_k$ , we now suppose that we can find a direction  $u_k$  belonging to the plane tangent at  $x_k$  to the face containing  $x_k$  and such that this direction is an approximate eigenvector corresponding to the most negative (leftmost) eigenvalue  $\tau_k$  of the reduced Hessian of the model Lagrangian. That is, we suppose that, for some  $z_k \in \mathbb{R}^{n-|\mathcal{A}(x_k)|}$ ,

$$u_k = N(x_k)z_k, \quad \langle P_{\mathcal{T}(x_k)}[-g_k], u_k \rangle \geq 0, \quad \|u_k\| = \Delta_k, \quad (12.4.12)$$

and

$$\langle u_k, \nabla_{xx} m_k^\ell(x_k) u_k \rangle = \left\langle u_k, \left[ \nabla_{xx} m_k(x_k) - \sum_{i=1}^m [y_k]_i \nabla_{xx} c_i(x_k) \right] u_k \right\rangle \leq \kappa_{\text{snc}} \tau_k \Delta_k^2 \quad (12.4.13)$$

for some  $\kappa_{\text{bck}} \in (0, 1)$  and  $\kappa_{\text{snc}} \in (0, 1]$ . We then define

$$t_j = \kappa_{\text{bck}}^j \quad \text{and} \quad x_k(j) = P_{\mathcal{C}}[x_k + t_j u_k]$$

for  $j \geq 0$  in the spirit of (6.6.8) (p. 150), except that we have to make sure (by using projections) that  $x_k(j)$  remains feasible. Applying, as before, the idea of backtracking, we then compute the smallest nonnegative integer  $j = j_e$  such that

$$m_k(x_k(j)) \leq m_k(x_k) - \kappa_{\text{ubc}} \tau_k t_j^2 \Delta_k^2, \quad (12.4.14)$$

where  $\kappa_{\text{ubc}} \in (0, \frac{1}{2} \kappa_{\text{snc}})$ . The *generalized eigenpoint* may then be defined by

$$t_k^{\text{GE}} = t_{j_e} \quad \text{and} \quad x_k^{\text{GE}} = x_k(t_k^{\text{GE}}) = P_{\mathcal{C}}[x_k + t_k^{\text{GE}} u_k].$$

Of course, we still have to verify that this point is well defined and to analyse what model reduction it guarantees. In order to obtain such results, we need further assumptions on the constraints functions, similar to those we made for the objective function in the unconstrained case.

**AC.4** There exists a constant<sup>195</sup>  $\kappa_{\text{uch}} > 0$  such that  $\|\nabla_{xx} c_i(x)\| \leq \kappa_{\text{uch}}$  for all  $i \in \{1, \dots, m\}$  and all  $x \in \mathcal{C}$ .

**AC.6** There exists a constant<sup>196</sup>  $\kappa_{\text{lcc}} > 0$  such that

$$\|\nabla_{xx} c_i(x) - \nabla_{xx} c_i(v)\| \leq \kappa_{\text{lcc}} \|x - v\|$$

for some  $\kappa_{\text{lcc}} > 0$ , all  $i \in \{1, \dots, m\}$ , and all  $x, v \in \mathcal{C}$ .

Note that AC.4 corresponds to AF.3, but for the constraints instead of the objective function, while AC.6 is the equivalent of AM.6 on the model. This apparent lack of symmetry results from our decision in this chapter to work directly with the constraints

<sup>195</sup> “uch” stands for “upper bound on the constraint Hessians”.

<sup>196</sup> “lcc” stands for “Lipschitz constant for the constraint Hessians”.

rather than by modelling them. We may interpret AC.4 as a guarantee that the curvature of the boundary of the feasible region along a face is bounded. AC.6 is nothing but Lipschitz continuity of the Hessians of the constraint functions.

**Theorem 12.4.6** Suppose that AF.1–AF.3, AC.1, AC.2, AC.4, AC.6, AC.7, AO.1c, AO.4b, AM.1–AM.3, AM.6, AA.6, and AI.1 hold. Suppose furthermore that the matrix  $N(x_k)^T \nabla_{xx} m_k^\ell(x_k) N(x_k)$  is indefinite and that  $\tau_k$  is its leftmost eigenvalue, and that  $u_k$  satisfies the conditions (12.4.12) and (12.4.13). Suppose in addition that there exist constants  $\kappa_{\text{ucj}} > 0$  and  $\kappa_{\text{ubm}} > 0$  independent of  $k$  such that

$$\|A(x_k)\| \leq \kappa_{\text{ucj}} \quad \text{and} \quad \|y_k\| \leq \kappa_{\text{ubm}}. \quad (12.4.15)$$

Then  $x_k^{\text{GE}}$  is well defined,

$$\sigma_k \stackrel{\text{def}}{=} \text{dist}(x_k, \partial \mathcal{F}(x_k)) > 0, \quad (12.4.16)$$

and there exists  $\kappa_{\text{sod}} > 0$  independent of  $k$  such that

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{sod}} \tau_k \min[1, \tau_k^2, \sigma_k^2, \Delta_k^2] \quad (12.4.17)$$

for both  $\|P_{\mathcal{T}(x_k)}[-g_k]\|$  and  $\|y_k - y_k^{\text{LS}}\|$  sufficiently small compared to  $-\tau_k$ .

**Proof.** Compare this proof with that of Theorem 6.6.2 (p. 151). We first observe that the assumption that  $N(x_k)^T \nabla_{xx} m_k^\ell(x_k) N(x_k)$  has a negative eigenvalue implies that the face,  $\mathcal{F}(x_k)$ , containing  $x_k$  cannot be a single point, because this would then imply that this reduced matrix is identically zero. Consequently, the face  $\mathcal{F}(x_k)$  is a proper (nontrivial) manifold in the neighbourhood of  $x_k$ , and (12.4.16) holds.

We then consider the case where (12.4.14) is violated for some  $j \geq 0$ . This yields that, for this  $j$ ,

$$m_k^\ell(x_k) - m_k^\ell(x_k(j)) = m_k(x_k) - m_k(x_k(j)) \leq -\kappa_{\text{ubc}} \tau_k t_j^2 \Delta_k^2, \quad (12.4.18)$$

where we used the relation (12.4.3) between  $m_k$  and  $m_k^\ell$  and the first part of AA.6 to deduce the first equality. We may now apply the mean value theorem to the first left-hand side of this relation and obtain that

$$\begin{aligned} & m_k^\ell(x_k) - m_k^\ell(x_k(j)) \\ &= -\langle \nabla_x m_k^\ell(x_k), x_k(j) - x_k \rangle - \frac{1}{2} \langle x_k(j) - x_k, \nabla_{xx} m_k^\ell(\xi_k)(x_k(j) - x_k) \rangle \\ &= \langle P_{\mathcal{T}(x_k)}[-g_k], x_k(j) - x_k \rangle + \langle y_k - y_k^{\text{LS}}, A(x_k)(x_k(j) - x_k) \rangle \\ &\quad - \frac{1}{2} \langle x_k(j) - x_k, \nabla_{xx} m_k^\ell(\xi_k)(x_k(j) - x_k) \rangle \end{aligned} \quad (12.4.19)$$

for some  $\xi_k \in [x_k, x_k(j)]$ , where we used the definition of  $m_k^\ell$ , AM.3, and the last three equalities of (12.4.1) to derive the last equality. We now define  $v_k(j) = x_k + t_j u_k$  and notice that  $x_k(j) = P_C[v_k(j)]$ . Using (12.4.18), (12.4.19), and the fact that  $A(x_k)u_k = 0$ , we then obtain that

$$\begin{aligned} & \kappa_{\text{ubc}} \tau_k t_j^2 \Delta_k^2 \\ & \leq -\langle P_{T(x_k)}[-g_k], x_k(j) - v_k(j) \rangle - t_j \langle P_{T(x_k)}[-g_k], u \rangle \\ & \quad + \langle y_k - y_k^{\text{LS}}, A(x_k)(x_k(j) - v_k(j)) \rangle \\ & \quad + \frac{1}{2} t_j^2 \langle u_k, \nabla_{xx} m_k^\ell(\xi_k) u_k \rangle + t_j \langle x_k(j) - v_k(j), \nabla_{xx} m_k^\ell(\xi_k) u_k \rangle \\ & \quad + \frac{1}{2} \langle x_k(j) - v_k(j), \nabla_{xx} m_k^\ell(\xi_k)(x_k(j) - v_k(j)) \rangle. \end{aligned} \quad (12.4.20)$$

Consider the fourth term on the right-hand side. Using (12.4.13) and the Cauchy–Schwarz inequality, we obtain that

$$\begin{aligned} & t_j^2 \langle u_k, \nabla_{xx} m_k^\ell(\xi_k) u_k \rangle \\ & = t_j^2 \langle u_k, \nabla_{xx} m_k^\ell(x_k) u_k \rangle + t_j^2 \langle u_k, [\nabla_{xx} m_k^\ell(\xi_k) - \nabla_{xx} m_k^\ell(x_k)] u_k \rangle \\ & \leq \kappa_{\text{snc}} \tau_k t_j^2 \Delta_k^2 + t_j^2 \Delta_k^2 \|\nabla_{xx} m_k^\ell(\xi_k) - \nabla_{xx} m_k^\ell(x_k)\|. \end{aligned}$$

From the definition of  $m_k^\ell$ , AM.6, and AC.6, we then immediately deduce, using the definition of  $\kappa_{\text{ubm}}$ , that

$$\begin{aligned} & \|\nabla_{xx} m_k^\ell(\xi_k) - \nabla_{xx} m_k^\ell(x_k)\| \\ & \leq \|\nabla_{xx} m_k(\xi_k) - \nabla_{xx} m_k(x_k)\| + \|y_k\| \max_i \|\nabla_{xx} c_i(\xi_k) - \nabla_{xx} c_i(x_k)\| \\ & \leq \kappa_{\text{lch}} \|\xi_k - x_k\| + \kappa_{\text{ubm}} \kappa_{\text{lcc}} \|\xi_k - x_k\|. \end{aligned}$$

But

$$\|\xi_k - x_k\| \leq \|x_k(j) - x_k\| \leq \|v_k(j) - x_k\| \leq t_j \Delta_k,$$

where we used the contractive nature of the projection and the identity  $x_k(j) = P_C[v_k(j)]$  to derive the second inequality. Hence,

$$t_j^2 \langle u_k, \nabla_{xx} m_k^\ell(\xi_k) u_k \rangle \leq \kappa_{\text{snc}} \tau_k t_j^2 \Delta_k^2 + [\kappa_{\text{lch}} + \kappa_{\text{ubm}} \kappa_{\text{lcc}}] t_j^3 \Delta_k^3.$$

Let us assume first that  $x_k(j) \in \mathcal{F}(x_k)$ . By the definition of  $x_k(j)$  as the projection of  $v_k(j)$  on the plane tangent to the constraints at  $x_k$ , we then have that

$$\|x_k(j) - v_k(j)\| \leq \kappa_{\text{uch}} \|v_k(j) - x_k\|^2 \leq \kappa_{\text{uch}} t_j^2 \Delta_k^2. \quad (12.4.21)$$

This situation is illustrated in Figure 12.4.1.

Finally, we observe that, because of AF.3, the definition of  $\kappa_{\text{ubm}}$ , and AC.4,

$$\|\nabla_{xx} m_k^\ell(\xi_k)\| \leq \|\nabla_{xx} m_k(\xi_k)\| + \|y_k\| \max_i \|\nabla_{xx} c_i(\xi_k)\| \leq \kappa_{\text{ufh}} + \kappa_{\text{ubm}} \kappa_{\text{uch}}.$$

Combining (12.4.20), all the above bounds, the Cauchy–Schwarz inequality, the

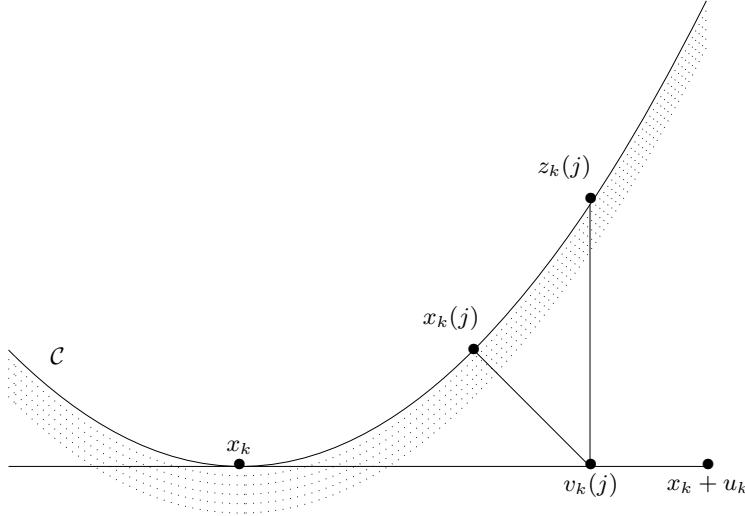


Figure 12.4.1: The first inequality of (12.4.21), seen in the plane  $(x_k, v_k(j), x_k(j))$ . One sees that  $\|x_k(j) - v_k(j)\| \leq \|z_k(j) - v_k(j)\| \leq \kappa_{\text{uch}} \|v_k(j) - x_k\|^2$ , where  $z_k(j)$  is the intersection of the constraint's boundary with the line perpendicular to  $u_k$  and containing  $v_k(j)$ .

definition of  $\kappa_{\text{ucj}}$ , and the second part of (12.4.12), we obtain that

$$\begin{aligned}
 & \kappa_{\text{ubc}} \tau_k t_j^2 \Delta_k^2 \\
 & \leq \|P_{T(x_k)}[-g_k]\| \|x_k(j) - v_k(j)\| + \|y_k - y_k^{\text{LS}}\| \|A(x_k)\| \|x_k(j) - v_k(j)\| \\
 & \quad + \frac{1}{2} t_j^2 \langle u_k, \nabla_{xx} m_k^\ell(\xi_k) u_k \rangle \\
 & \quad + t_j \Delta_k \|x_k(j) - v_k(j)\| \|\nabla_{xx} m_k^\ell(\xi_k)\| + \frac{1}{2} \|x_k(j) - v_k(j)\|^2 \|\nabla_{xx} m_k^\ell(\xi_k)\| \\
 & \leq \kappa_{\text{uch}} [\|P_{T(x_k)}[-g_k]\| + \kappa_{\text{ucj}} \|y_k - y_k^{\text{LS}}\|] t_j^2 \Delta_k^2 + \frac{1}{2} \kappa_{\text{snc}} \tau_k t_j^2 \Delta_k^2 \\
 & \quad + \frac{1}{2} [\kappa_{\text{lch}} + \kappa_{\text{ubm}} \kappa_{\text{lcc}}] t_j^3 \Delta_k^3 + [\kappa_{\text{uch}} t_j^3 \Delta_k^3 + \frac{1}{2} \kappa_{\text{uch}}^2 t_j^4 \Delta_k^4] [\kappa_{\text{ufh}} + \kappa_{\text{ubm}} \kappa_{\text{uch}}]. \tag{12.4.22}
 \end{aligned}$$

We now use our assumption on  $\|P_{T(x_k)}[-g_k]\|$  and  $\|y_k - y_k^{\text{LS}}\|$  and assume that

$$\|P_{T(x_k)}[-g_k]\| + \kappa_{\text{ucj}} \|y_k - y_k^{\text{LS}}\| \leq \frac{\kappa_{\text{ubc}} - \frac{1}{2} \kappa_{\text{snc}}}{2 \kappa_{\text{uch}}} \tau_k, \tag{12.4.23}$$

where the right-hand side of this last condition is positive because of the bound  $\kappa_{\text{ubc}} < \frac{1}{2} \kappa_{\text{snc}}$ . Then, if

$$t_j \Delta_k < 1, \tag{12.4.24}$$

we finally obtain, after dividing (12.4.22) by  $t_j^2 \Delta_k^2$ , that

$$\kappa_{\text{ubc}} \tau_k \leq \frac{1}{2} [\kappa_{\text{ubc}} + \frac{1}{2} \kappa_{\text{snc}}] \tau_k + \kappa_0 t_j \Delta_k,$$

where we have set

$$\kappa_0 = \frac{1}{2} [\kappa_{\text{lch}} + \kappa_{\text{ubm}} \kappa_{\text{lcc}}] + [\kappa_{\text{uch}} + \frac{1}{2} \kappa_{\text{uch}}^2] [\kappa_{\text{ufh}} + \kappa_{\text{ubm}} \kappa_{\text{uch}}].$$

This in turn yields that

$$t_j \Delta_k \geq \frac{\kappa_{\text{ubc}} - \frac{1}{2}\kappa_{\text{snc}}}{2\kappa_0} \tau_k.$$

If, on the other hand, (12.4.24) fails, then we obtain that  $t_j \Delta_k \geq 1$ . In both cases, we thus deduce that

$$t_j \Delta_k \geq \min \left[ 1, \frac{\kappa_{\text{ubc}} - \frac{1}{2}\kappa_{\text{snc}}}{2\kappa_0} \tau_k \right] > 0. \quad (12.4.25)$$

But  $t_j$  tends to zero when  $j$  tends to infinity, and thus (12.4.25) implies that (12.4.18) must be false for  $j$  sufficiently large, provided (12.4.23) holds. As a consequence, there must exist a finite smallest  $j_e$  such that (12.4.14) holds, which shows that  $j_e$  and therefore  $x_k^{\text{GE}}$  are well defined. Moreover, (12.4.14) implies that

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{ubc}} - \tau_k t_{j_e}^2 \Delta_k^2. \quad (12.4.26)$$

We then deduce from

$$t_{j_e} = \kappa_{\text{bck}} t_{j_e-1}$$

and the fact that (12.4.26) and hence the first part of (12.4.25) hold for  $j = j_e - 1$  that

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{ubc}} \kappa_{\text{bck}}^2 \min \left[ \left[ \frac{\kappa_{\text{ubc}} - \frac{1}{2}\kappa_{\text{snc}}}{2\kappa_0} \right]^2 \tau_k^2, 1 \right] - \tau_k. \quad (12.4.27)$$

Now consider the sequence  $\{x_k(j)\}$ . Because it converges to  $x_k$  and because we have already seen that  $\sigma_k > 0$ , there must be a smallest  $j_i$  such that  $x_k(j_i) \in \mathcal{F}(x_k)$ . Then either (12.4.25) holds with  $j = j_i$ , from which we deduce that  $j_e > j_i$  and (12.4.27) holds, or (12.4.25) fails for  $j = j_i$ , and thus (12.4.18) must be false for  $j = j_i$  provided (12.4.23) holds. This implies that  $x_k^{\text{GE}} = x_k(j_i)$  and that

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{ubc}} |\tau_k| t_{j_i}^2 \Delta_k^2.$$

By the definition of  $j_i$  we obtain that

$$\|x_k(j_i - 1) - x_k\| \geq \sigma_k$$

and thus that

$$t_{j_i} \Delta_k = \kappa_{\text{bck}} t_{j_i-1} \Delta_k \geq \kappa_{\text{bck}} \|x_k(j_i - 1) - x_k\| \geq \kappa_{\text{bck}} \sigma_k.$$

Hence we deduce that, in this case,

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{ubc}} \kappa_{\text{bck}} - \tau_k \sigma_k^2. \quad (12.4.28)$$

We finally turn to the case where (12.4.14) holds for all  $j \geq 0$ , in which case  $j_e = 0$  and  $x_k^{\text{GE}}$  is again well defined. In that case,  $t_{j_e} = 1$  and (12.4.14) gives that

$$m_k(x_k) - m_k(x_k^{\text{GE}}) \geq \kappa_{\text{ubc}} - \tau_k \Delta_k^2.$$

Combining this last bound with (12.4.27) and (12.4.28), we obtain (12.4.17) with

$$\kappa_{\text{sod}} \stackrel{\text{def}}{=} \kappa_{\text{ubc}} \min \left[ \kappa_{\text{bck}}^2 \left[ \frac{\kappa_{\text{ubc}} - \frac{1}{2}\kappa_{\text{snc}}}{\kappa_{\text{lch}} + \kappa_{\text{ubm}}\kappa_{\text{lcc}} + [2\kappa_{\text{uch}} + \kappa_{\text{uch}}^2][\kappa_{\text{ufh}} + \kappa_{\text{ubm}}\kappa_{\text{uch}}]} \right]^2, \kappa_{\text{bck}} \right]$$

since  $\kappa_{\text{bck}} < 1$ .  $\square$

The bound (12.4.17) is a little more complex than what we already saw in AA.2 or AA.2b, but we will see that it is still adequate. Observe also that our assumption on the existence of  $\kappa_{\text{ucj}}$  is unnecessary if  $y_k = y_k^{\text{LS}}$ , since all the terms involving the Jacobian matrix  $A(x_k)$  then vanish. If, as in the unconstrained case, we require the model reduction at the trial step to be at least (a fraction of) that obtained at the generalized eigenpoint, we see that we may, in our context, reasonably assume the following variant of AA.2/AA.2b.

**AA.2c** If  $\tau_k = \lambda_{\min} [N(x_k)^T \nabla_{xx} m_k^\ell(x_k) N(x_k)] < 0$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} - \tau_k \min[1, \tau_k^2, \sigma_k^2, \Delta_k^2]$$

for some constant  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$ , when  $\|y_k - y_k^{\text{LS}}\|$  and  $\|P_{\mathcal{T}(x_k)}[-g_k]\|$  are sufficiently small compared to  $-\tau_k$ .

We may then extend the results of Theorems 6.6.3–6.6.5, 6.6.7, and 6.6.8 to our constrained framework, under the further restriction that all iterates  $\{x_k\}$  lie within a closed, bounded domain (AI.1). With the help of this additional assumption, our development then follows the outline of Section 6.6.3 very closely if one takes the following three considerations into account.

The first is that AI.1 not only allows us to apply the constraint identification result of Theorem 12.3.8 but also ensures that the sequence  $\{x_k\}$  may be partitioned into a number of converging subsequences. Hence Theorem 12.3.3 yields that all iterates belong to  $\mathcal{V}(\mathcal{L}_*, \delta)$  for some  $\delta > 0$  and  $k$  sufficiently large. This has two interesting consequences. First, the continuity of the constraints gradients (AC.1) on  $\mathcal{V}(\mathcal{L}_*, \delta)$  then implies that the first part of (12.4.15) holds for some  $\kappa_{\text{ucj}} > 0$ . Furthermore, the continuity of the gradient of the objective function (AF.1), the boundedness of  $\{x_k\}$ , and AM.3 ensure that  $g_* = \nabla_x f(x_*)$  is finite and thus, because of AO.1c, that  $y_*$  is well defined and finite. We then deduce from (12.4.11) and AA.6 that the multipliers  $y_k$  must also be finite, which ensures that the second part of (12.4.15) holds as well. A second consequence of AI.1 is that we may restrict our attention to the analysis of the limit points of convergent subsequences of iterates.<sup>197</sup> In particular, we may associate to each limit point  $x_*$  a value

$$\sigma_* = \text{dist}(x_*, \partial \mathcal{F}(x_*)) > 0,$$

and we thus obtain that  $\sigma_k \geq \frac{1}{2}\sigma_* > 0$  for each iterate  $x_k$  that is sufficiently close to  $x_*$ . Thus, for such an  $x_k$ , the bound in AA.2c reduces to that in AA.2b with

---

<sup>197</sup>Note that Theorem 6.6.4 (p. 154) is not meaningful outside the context of Theorem 6.6.5 (p. 155).

$\kappa_{\text{lsd}} = \min[1, \frac{1}{2}\sigma_*]$ . As was the case for AA.2b in Section 8.1.3, these constants turn out to be irrelevant because one always has that  $\Delta_k$  tends to zero when one calls upon the properties of the (generalized) eigenpoint, and the last term in the minimum therefore always dominates.

The second observation is that one is no longer interested in the eigenstructure of  $\nabla_{xx}f(x_*)$ , but rather in that of  $N(x_*)^T \nabla_{xx}\ell(x_*, y_*)N(x_*)$ . As indicated above, AM.5b and AA.6 together imply the convergence of the Hessian of the model, that is,

$$\lim_{i \rightarrow \infty} \left[ \nabla_{xx}m_{k_i}(x_{k_i}) - \sum_{i=1}^m [y_{k_i}]_i \nabla_{xx}c_i(x_{k_i}) \right] = \nabla_{xx}f(x_*) - \sum_{i=1}^m [y_*]_i \nabla_{xx}c_i(x_*),$$

and therefore that

$$\begin{aligned} & \lim_{i \rightarrow \infty} N(x_{k_i})^T \left[ \nabla_{xx}m_{k_i}(x_{k_i}) - \sum_{i=1}^m [y_{k_i}]_i \nabla_{xx}c_i(x_{k_i}) \right] N(x_{k_i}) \\ &= N(x_*)^T \left[ \nabla_{xx}f(x_*) - \sum_{i=1}^m [y_*]_i \nabla_{xx}c_i(x_*) \right] N(x_*), \end{aligned} \quad (12.4.29)$$

which is the equivalent of AM.5b in the unconstrained context.<sup>198</sup> Thus, assuming that the matrix on the right-hand side of this last relation has a (most) negative eigenvalue  $\lambda_*$ , we may assume that, for  $k$  large enough, the matrix under the limit on the left-hand side has an eigenvalue that is at most  $\frac{1}{2}\lambda_*$ . This argument is used in the beginning of the proofs of Theorem 6.6.4 (p. 154)/8.1.2, Lemma 6.6.6 (p. 156), and Theorems 6.6.7 (p. 157) and 6.6.8 (p. 159)/8.1.5.

The third and last consideration is that Theorem 12.4.6, at variance with its unconstrained counterpart (Theorem 6.6.2 [p. 151]), also requires that the projected negative gradient and the difference between the multipliers and their least-squares version must be small, which is reflected in the statement of AA.2c. Fortunately, this has no consequence in the proofs of Theorems 6.6.3–6.6.5, 6.6.7, and 6.6.8 or their versions of Section 8.1.3, since AA.2c is always called upon in the case where these conditions are satisfied. Indeed, the theorem is always applied for  $k$  sufficiently large, which ensures, because of Corollary 12.3.9 and AA.6, that the size of the projected negative gradient,  $\|P_{\mathcal{T}(x_k)}[-g_k]\|$ , and of the error in the multipliers,  $\|y_k - y_k^{\text{LS}}\|$ , must be as small as required. They must therefore become sufficiently small compared to the absolute value of the (assumed) negative eigenvalue  $\frac{1}{2}\lambda_*$ .

Thus we have obtained that all *the conclusions of Theorems 6.6.3, 6.6.5, 6.6.7, and 6.6.8 are still valid* for Algorithm 12.2.1 if we assume AF.1–AF.3, AC.1, AC.2, AC.4, AC.6, AC.7, AO.1c, AO.4b, AM.1–AM.4, AM.5b, AM.6, AA.1b, AA.2c, AA.3 (possibly), AA.5, AA.6, and AI.1. Hence all limit points of the sequence of iterates produced by Algorithm 12.2.1 are second-order critical, but it is important to note that only the weak second-order necessary conditions are satisfied as these limit points. This is because we have restricted our search for the generalized eigenpoint to the projection of a negative curvature direction  $u_k$  belonging to the plane tangent to the active constraints. Perhaps we could replace the first part of (12.4.12) by the requirement that

---

<sup>198</sup>Observe that we could only require (12.4.29) instead of AM.5b and AA.6, which would be weaker.

$u_k$  belong to the tangent cone at  $x_k$  and correspondingly obtain strong second-order conditions at the limit points. However, this stronger result would assume that we are able to find such a negative curvature direction in the tangent cone whenever it exists, which is, unfortunately, a difficult computational problem. For instance, even for the simplest possible case, where the only constraints are that the variables must be nonnegative, this amounts to deciding if the Hessian of the Lagrangian is “copositive”, a task which is NP complete.<sup>199</sup>

## Notes and References for Section 12.4

For the case of convex models, we have followed the outline of Chapter 6, Section 6.5, but not in its entirety. The attentive reader will have noticed that the assumptions of Theorem 12.4.3 are slightly more restrictive than those of Theorem 6.5.2 (p. 141), even despite the fact that constraints are now present. Indeed, we have assumed convexity of the Lagrangian in the tangent space, instead of the mere nonsingularity of its Hessian, which would be the more direct extension of the unconstrained result. However, it is also interesting to note that this theorem is more general than that given by Conn et al. (1993, Theorem 6.1), where convexity of the objective function, and not just the Lagrangian, is required. On the other hand, Conn et al. cover the case where approximate derivatives (in the sense of Section 8.4) are used, a level of generality that we have avoided here to improve readability.

The theory of convergence to second-order critical points using the nonconvex model is new, although directly inspired by the developments of Section 6.6. Branch (1995) also discusses the behaviour of an algorithm for bound-constrained problems in the presence of negative curvature directions.

Conn et al. (1993) also consider the case where computing the projection of a point onto the feasible region may not be as easy as assumed in this chapter. In this case, it is possible to relax the conditions defining the generalized Cauchy point via an “inexact projection” in such a way that it need no longer be on the projected-gradient path, but is simply required to achieve a fraction of the decrease in the linearized objective that would be obtained by a point on this path. The projection problem (involving a quadratic objective function) is then replaced by a minimization problem with a linear objective, which may be computationally more tractable, especially when the constraints are also linear. In this case, it is necessary to modify Algorithm 12.2.2 to allow for this approximation, while still guaranteeing that the generalized Cauchy point remains well defined, ensures a sufficient model reduction, and ultimately converges.

---

<sup>199</sup>See, for example, Murty and Kabadi (1987).

# Chapter 13

---

## Barrier Methods for Inequality Constraints

---

### 13.1 Convex Constraints and Barriers

In the previous chapter, we considered our first important class of methods for solving constrained optimization problems, specifically those that proceed by projection onto the feasible region. In this chapter, we consider an alternative. Like its predecessor, all the iterates generated will be feasible. But unlike the methods in Chapter 12, this will be ensured implicitly rather than explicitly. In particular, the iterates of our previous methods “hug” the boundary of the feasible region—recall, we showed that under perhaps idealized conditions, the set of constraints active at a limit point was accurately predicted by those that are active well before the limit is reached. By contrast, the methods we shall now consider approach their limit points from the strict interior of the feasible region. They are prevented from crossing (or even moving to) the boundary of the feasible region by the very act of minimization. To achieve this, we shall construct a parametrized, artificial objective function (known as a barrier function) that is finite within the feasible region—and can be made to be an increasingly good approximation to the original objective there simply by adjusting the parameters—but that approaches infinity whenever an unwary iterate tries to move into infeasibility.

We start by considering the very simple case with the feasible region  $\mathcal{C}$  given as

$$\mathcal{C} = \mathcal{O}_+ \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid x \geq 0\},$$

where the inequality  $x \geq 0$  holds componentwise. In other words, we restrict the minimization of our objective function to nonnegative variables. We may then consider the following *logarithmic barrier function*<sup>200</sup>

$$\phi^{\log}(x, \mu) \stackrel{\text{def}}{=} f(x) - \mu \langle e, \log(x) \rangle = f(x) - \mu \sum_{i=1}^n \log[x]_i, \quad (13.1.1)$$

---

<sup>200</sup>Often called the log-barrier.

where  $\mu > 0$  is called the *barrier parameter*,  $\log(x)$  is the vector whose  $i$ th component is the logarithm of the  $i$ th component of  $x$ , and

$$b^{\log}(x, \mu) = -\mu \langle e, \log(x) \rangle \quad (13.1.2)$$

is the *log-barrier term*. Because of the kind of singularity of the logarithm at zero, the function  $\phi^{\log}(x, \mu)$  will be finite on the interior of the positive orthant ( $\mathcal{C}$ ), infinitely large on its boundary  $\partial\mathcal{C}$ , and undefined outside. The idea is then to solve a sequence of problems of the form

$$\min_x \phi^{\log}(x, \mu)$$

parametrized by smaller and smaller values of  $\mu$ . Under reasonable assumptions, we will see that the solutions

$$x_*(\mu) = \arg \min_x \phi^{\log}(x, \mu)$$

converge to the solution of the problem

$$\min_{x \in \mathcal{C}} f(x) \quad (13.1.3)$$

when  $\mu$  converges to zero. More importantly, we shall show that, under similar assumptions, first-order critical points for the former converge to first-order critical points for the latter as  $\mu$  approaches zero.

The process is illustrated in Figure 13.1.1, where we consider the problem

$$\min_{x_1, x_2 \geq 0} 120 \left[ x_1^2(x_1 - 1) - x_2 + 1 \right]^2 + 10(4 + x_1)^2 - 150. \quad (13.1.4)$$

The contour lines of the objective function in the positive orthant are shown together with those of the corresponding log-barrier function for five values of the barrier parameter. The increasing values of  $\phi^{\log}(x, \mu)$  as  $x$  approaches the boundary of  $\mathcal{C}$  explains the epithet “barrier function”, since  $x$  will be prevented from becoming zero or negative if a descent method is used to minimize  $\phi^{\log}$  starting from an interior point.

Having illustrated the basic ideas behind barrier function methods, we now turn to the slightly more general context in which we consider a general barrier term  $b(x, \mu)$ , the associated barrier function

$$\phi(x, \mu) = f(x) + b(x, \mu),$$

and a general convex feasible region  $\mathcal{C}$ . We shall therefore assume that

**AC.2b** the feasible region has a nonempty relative interior; that is,  $\text{ri}\{\mathcal{C}\} \neq \emptyset$ .

In order to make useful assumptions about the barrier term, we have to decide on a norm to use in the space of variables. For the first part of this chapter, we will make the simplest and most intuitive choice possible, by picking the  $\ell_2$  norm for our analysis. However, as we will see in Section 13.7, other choices are possible and, in fact, highly desirable for numerical efficiency. We will therefore return to the more general framework of iteration-dependent norms at that point. In the meantime, we let

$$\text{dist}(y, \mathcal{Z}) \stackrel{\text{def}}{=} \inf_{z \in \mathcal{Z}} \|y - z\|$$

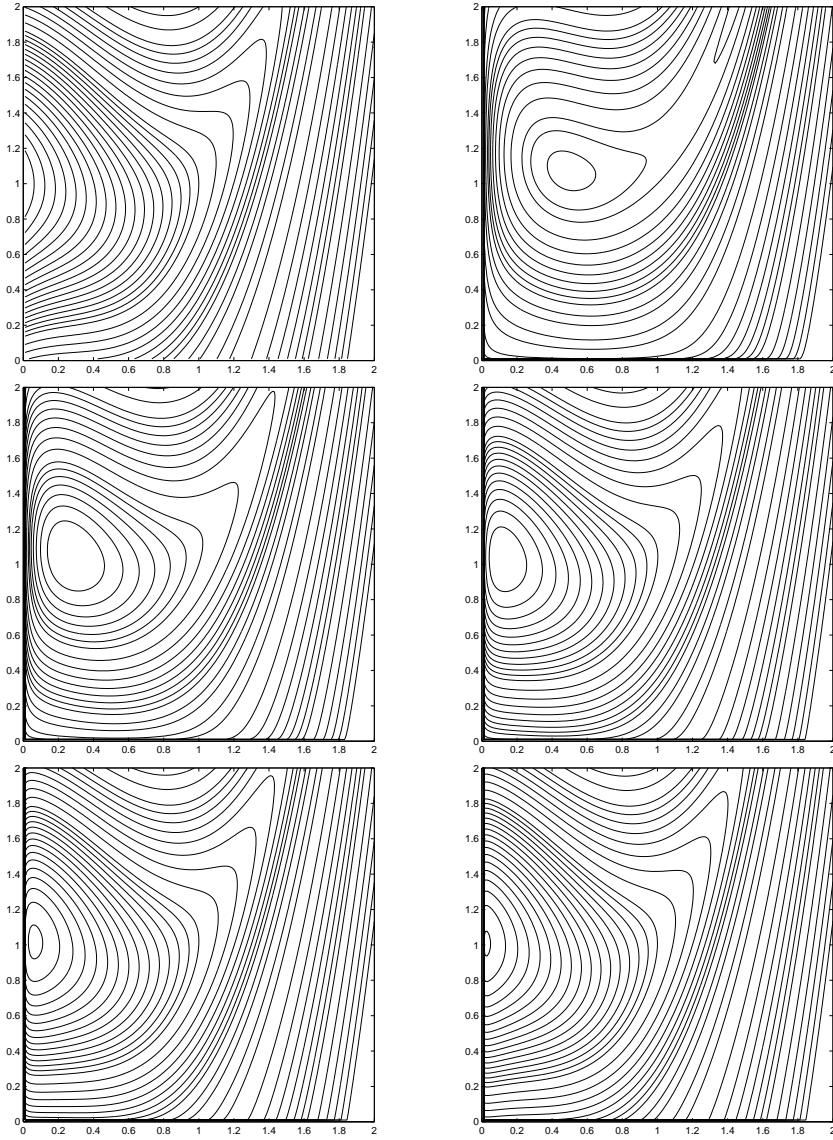


Figure 13.1.1: A simple minimization problem in the positive orthant: contours of the objective function (top left) and of the log-barrier function for  $\mu = 50$  (top right),  $\mu = 25$  (middle left),  $\mu = 10$  (middle right),  $\mu = 5$  (bottom left), and  $\mu = 2$  (bottom right). The contour lines of the log-barrier function for  $\mu = 0.1$  are almost visually indistinguishable at this scale from those of the original problem.

measure the distance from the point  $y$  to the set  $\mathcal{Z} \subseteq \mathbb{R}^n$ . Note that the triangle inequality immediately implies that

$$\text{dist}(x + y, \partial\mathcal{C}) \leq \text{dist}(x, \partial\mathcal{C}) + \|y\|. \quad (13.1.5)$$

The barrier term  $b(x, \mu)$  is then required to satisfy the following conditions.

**AC.1c** The barrier term  $b(x, \mu)$  is defined for all  $x \in \text{ri}\{\mathcal{C}\}$  and all  $\mu > 0$ , and is twice-continuously differentiable on  $\text{ri}\{\mathcal{C}\}$  with respect to its first argument.

**AC.4c** For each  $\mu > 0$  and each  $\epsilon > 0$ , there exists a constant<sup>201</sup>  $\kappa_{\text{bbh}}(\epsilon, \mu) \geq 1$  such that

$$\|\nabla_{xx} b(x, \mu)\| \leq \kappa_{\text{bbh}}(\epsilon, \mu)$$

for all  $x \in \mathcal{C}$  such that  $\text{dist}(x, \partial\mathcal{C}) \geq \epsilon$ .

**AC.8** One has that

$$\lim_{p \rightarrow \infty} b(y_p, \mu) = +\infty$$

for any  $\mu > 0$  and any sequence  $\{y_p\}_{p=0}^{\infty}$  such that

$$y_p \in \text{ri}\{\mathcal{C}\} \text{ for all } p \text{ and } \lim_{p \rightarrow \infty} \text{dist}(y_p, \partial\mathcal{C}) = 0.$$

We defer a discussion of how reasonable these assumptions are until we consider specific barrier terms. It is enough to mention here (without further justification) that they are satisfied for (13.1.2).

The following iteration then defines a typical barrier algorithm.

**Algorithm 13.1.1: General barrier algorithm**

**Step 0: Initialization.** An initial point  $x_0 \in \text{ri}\{\mathcal{C}\}$  and an initial barrier parameter  $\mu_0 > 0$  are given. Set  $k = 0$ .

**Step 1: Inner minimization.** Solve (approximately) the problem

$$\min_x \phi(x, \mu_k) \tag{13.1.6}$$

by applying an unconstrained minimization algorithm, starting from a suitable starting point  $x_{k,0} \in \text{ri}\{\mathcal{C}\}$ . Let  $x_{k+1}$  be the corresponding (approximate) solution.

**Step 2: Update the barrier parameter.** Choose  $\mu_{k+1} > 0$  in such a way as to ensure that

$$\lim_{k \rightarrow \infty} \mu_k = 0.$$

Increment  $k$  by 1 and return to Step 1.

---

<sup>201</sup>“bbh” stands for “bound on the barrier Hessian”.

We let the sequence of iterates generated by the minimization in Step 1 be  $\{x_{k,j}\}$  ( $j \geq 0$ )—these are called *inner iterates*, and the corresponding iteration of the unconstrained minimization algorithm is the *inner iteration*. We have been deliberately vague about how the starting point for the inner iterations,  $x_{k,0}$ , is obtained. A simple technique is to choose

$$x_{k,0} = x_k,$$

but this is not the only possible one. In fact, we would like to choose  $x_{k,0}$  such that the number of inner iterations is as small as possible (ideally one). The inner minimization process terminates with

$$x_{k+1} = x_{k,j}$$

for some  $j > 0$ ,  $x_{k,j}$  being an approximate solution of problem (13.1.6). The iteration covering Steps 1 and 2 (and indexed by  $k$ ) is the *outer* iteration, and the sequence  $\{x_k\}$  is the sequence of outer iterates. To analyse a barrier method, one therefore needs to consider the behaviour of both the inner and outer iterations.

## Notes and References for Section 13.1

The logarithmic-barrier function method for finding a local minimizer of a nonlinear objective function subject to a set of inequality constraints was first introduced by Frisch (1954, 1955). The method was put on a sound theoretical framework by Fiacco and McCormick (1968), who also provided an interesting history of such techniques up until then. These authors showed that, under extremely modest conditions, the sequence of inner minimizers converges to the solution of the original problem whenever the sequence of barrier parameters converges to zero. In particular, under a strict complementary slackness assumption, the difference between the minimizer of the inner problem and the solution to the original problem is of order  $\mu$  as  $\mu$  tends to zero. (Mifflin, 1975a, shows an order  $\sqrt{\mu}$  error in the absence of the complementary slackness assumption and a weakening of the assumption that the inner problem be solved exactly.) For further discussion, see the survey by Wright (1992) and the contributions by M. H. Wright (1998) and Wright and Orban (1999).

It was originally envisaged that each of the sequence of barrier functions be minimized using standard methods for unconstrained minimization. However, Lootsma (1969) and Murray (1971b) painted a less optimistic picture by showing that, under most circumstances, the spectral condition number of the Hessian matrix of the barrier function increases without bound as  $\mu$  shrinks. This has important repercussions, as it suggests that a simple-minded sequential minimization is likely to encounter numerical difficulties. Consequently, the initial enthusiasm for barrier function methods declined. Methods that directly aim to alleviate these difficulties for smaller, dense problems have been proposed by Murray (1969), Wright (1976), Murray and Wright (1978), and McCormick (1991), while methods that are equally applicable to large, sparse problems have also been suggested (see, for instance, Gould, 1986, and Nash and Sofer, 1993).

Interest in the use of barrier functions was rekindled by the seminal paper of Karmarkar (1984) on polynomial time interior-point algorithms for linear programming and by the intimate connection between these methods and barrier function methods observed by Gill et al. (1986). The ill-conditioning problems described above do *not* occur for primal and dual

nondegenerate linear programs, as the solutions to such problems occur at vertices of the constraint boundary. Furthermore, even in the presence of degeneracy, stable numerical methods can be used; see Murray (1992), Fourer and Mehrotra (1993), and Andersen et al. (1996). Moreover, and most significantly, these methods have turned out to be most effective in practice for linear programming. We refer the reader to the excellent book on the subject by Wright (1997).

Remarkably, although the ill-conditioning difficulties are present in most nonlinear programs, the effects may be benign provided sufficient care is taken. In particular, Ponceleón (1990) has shown that if the only constraints that are handled by logarithmic terms are simple bounds, the ill-conditioning manifests itself solely on the diagonal of the Hessian matrix of the barrier function. She then shows, by a sensitivity analysis, that such terms are ultimately irrelevant in assessing the sensitivity of the Newton equations for the problem to numerical perturbations in the data. Methods of this sort have been successfully applied to the minimization of nonlinear functions whose only constraints are simple bounds on the variables. See, for instance, Nash and Sofer (1993) as well as the primal-dual method and Coleman and Li's method described later in this chapter. More significantly, S. J. Wright (1998) and M. H. Wright (1999) showed that in general the ill-conditioning is not harmful, as the potential large errors are confined to subspaces where they do little damage. This is due almost entirely to the special form of the right-hand side of the Newton equations and means that almost any (backward) stable factorization will produce accurate solutions to the Newton equations for all but the smallest values of  $\mu$ . The main implication is that the previously mentioned techniques for avoiding the ill-conditioning may not be necessary after all.

A further development of barrier methods aims at avoiding the ill-conditioning problem by other means. To do this, several authors have suggested modifying the definition of the barrier term. To our knowledge, the first move in this direction was the work by Jittorntrum and Osborne (1980) for the problem of minimizing  $f(x)$  subject to the constraints  $c_i(x) \geq 0$  ( $i = 1, \dots, m$ ), using a sequential minimization of a variant of the log-barrier function given by

$$f(x) - \mu \sum_{i=1}^m y_i \log(c_i(x))$$

for appropriate Lagrange multiplier estimates  $y_i$ . They showed that it is possible to get better than linear error estimates of the solution as  $\mu$  converges to zero merely by choosing the Lagrange multiplier estimates carefully. In the same spirit, we also mention the “shifted-barrier” method, which was analysed for linear programs by Gill et al. (1988), and the class of “modified barrier” methods proposed by Polyak (1982) and analysed in Polyak (1992). Gill et al. consider the *shifted* barrier function

$$f(x) - \sum_{i=1}^m w_i \log(c_i(x) + s_i),$$

where the  $w_i$  are termed *weights* and the  $s_i$  called *shifts*. A sequence of shifted barrier functions is minimized subject to the restriction that the ratios  $w_i/s_i$  converge to the Lagrange multipliers associated with the solution of the original problem. The authors prove convergence of such a scheme under mild conditions for linear programming problems. Polyak (1982) considers the *modified* barrier function

$$f(x) - \mu \sum_{i=1}^m y_i \log\left(1 + \frac{c_i(x)}{\mu}\right), \quad (13.1.7)$$

which he motivates by noting the equivalence of the constraints  $c_i(x) \geq 0$  and

$$\mu \log \left( 1 + \frac{c_i(x)}{\mu} \right) \geq 0 \text{ for } i = 1, \dots, m. \quad (13.1.8)$$

The function (13.1.7) is then merely the classical Lagrangian function for the problem of minimizing  $f(x)$  subject to the constraints (13.1.8). Polyak shows that, provided  $\mu$  is sufficiently small and other reasonable assumptions are satisfied, a sequential minimization of (13.1.7), in which  $\mu$  remains fixed but the Lagrange multipliers are adjusted, will converge to a solution of the original problem. This has the desirable effect of limiting the size of the condition number of the Hessian matrix of (13.1.7). Breitfeld and Shanno (1994) point out that this additional flexibility allows equality constraints to be handled via two (shifted) inequalities. Finally, Freund (1991), Jensen, Polyak, and Schneur (1992), and Powell (1993) have analysed and implemented shifted and modified barrier function methods for linear programming. Jensen and Polyak (1994) extend this work to convex problems. Further computational experience is reported by Breitfeld and Shanno (1996). In the same class of algorithms, we finally mention the *Lagrangian* barrier method of Conn, Gould, and Toint (1997a), where the “Lagrangian barrier” function

$$f(x) - \sum_{i=1}^m y_i s_i \log(c_i(x) + s_i)$$

is considered and for which global and fast linear convergence is proved. Without giving details, we note that this latter method corresponds very closely to an augmented Lagrangian method<sup>202</sup> for the original problem when the constraints are reformulated as

$$s_i \log \left( 1 + \frac{c_i(x)}{s_i} \right) \geq 0 \text{ for } i = 1, \dots, m.$$

Further work on this method can be found in Conn, Gould, and Toint (1994, 1997c). Although they are most effective for some classes of applications (such as structural analysis; see Ben-Tal and Zibulevsky, 1997), it is our experience that all of these variants of the barrier method suffer from a high sensitivity of their performance to the choice of parameters. This restricts their use as general-purpose methods and is the reason why we shall not explore them further in this book.

We conclude this brief historical survey of barrier function methods by mentioning an important and relatively recent development for the case of convex problems: *self-concordant barriers*. This general class of thrice-continuously differentiable barrier terms was first introduced and studied by Nesterov and Nemirovskii (1994). The formal definition of these barrier terms ensures that they are not “too nonlinear” in that their gradients and third derivative tensors are bounded in terms of their Hessians. They are of interest because it can be proved that Newton’s method is especially efficient when applied to such barriers, even without globalizing techniques such as trust regions. Other useful sources of information pertaining to self-concordant barriers include the papers Jarre and Saunders (1995) and Nesterov and Todd (1998), but we should warn the reader that the accumulating literature on this and related areas is vast. Interestingly, the logarithmic barrier term (13.1.2) that we will use throughout this chapter turns out to be self-concordant for the positive orthant, and this is also the case for its direct generalization to convex polytopes. But the class includes less obvious cases, such as the barrier defined for the cone  $\mathcal{S}_n$  of positive semidefinite symmetric matrices which is given by

$$b(X, \mu) = -\log [\det(X)]$$

---

<sup>202</sup>See Chapter 14 for a more detailed discussion of augmented Lagrangian techniques.

(where  $X$  is now a symmetric matrix). The discovery of this barrier spawned the whole area of *semidefinite programming* whose purpose is to minimize convex functions on the cone  $\mathcal{S}_n$ . Developments in this area are reviewed in Vandenberghe and Boyd (1996) and Fujisawa, Kojima, and Nakata (1997), but the interested reader should also consult Boyd et al. (1994) for applications in control theory, Ben-Tal and Nemirovskii (1997) for structural optimization problems, or Alizadeh (1995) and Goemans (1997) for the use of semidefinite programming in combinatorial optimization. However, since the focus of this book is on globally convergent algorithms for nonconvex problems, we choose not to consider self-concordant barriers in more detail.

## 13.2 A Trust-Region Method for Barrier Functions

In order to solve the problem (13.1.6) (the inner minimization problem), the obvious approach is to apply a standard trust-region algorithm like Algorithm BTR (p. 116). However, the fact that  $b(x, \mu)$  is undefined wherever one of the  $x_{k,j}$  does not belong to  $\text{ri}\{\mathcal{C}\}$  creates a difficulty, for nothing in the theory given in Chapter 6 prevents the model at the inner iteration  $j$  from predicting a step  $s_{k,j}$  such that  $x_{k,j} + s_{k,j} \notin \text{ri}\{\mathcal{C}\}$ . The value  $b(x_{k,j} + s_{k,j}, \mu_k)$ , and therefore  $\phi(x_{k,j} + s_{k,j}, \mu_k)$ , are then undefined, and Algorithm BTR in principle breaks down. Fortunately, such undesirable behaviour can easily be circumvented. We discuss the necessary modification of Algorithm BTR in this section. We remind the reader that, for simplicity of exposition, we shall temporarily let  $\|\cdot\|_{k,j} = \|\cdot\|$  for all  $k$  and  $j$ .

The idea is to observe that, if  $x_{k,j} + s_{k,j}$  lies outside  $\mathcal{C}$ , this is merely an indication that the model  $m_{k,j}$  does not approximate the objective  $\phi(x_{k,j} + s, \mu_k)$  very well. In particular, this indicates that a smaller step from  $x_{k,j}$  (which must lie inside  $\mathcal{C}$ ) is necessary. The first simple idea is to restrict the trust-region radius enough to ensure that

$$x_{k,j} + s_{k,j} \in \text{ri}\{\mathcal{C}\},$$

which must occur when  $\Delta_k$  is small enough to guarantee that

$$\mathcal{B}_{k,j} \subset \text{ri}\{\mathcal{C}\}.$$

The crucial point is that this restriction may be decided without even trying to compute the (undefined) function value at  $x_{k,j} + s_{k,j}$ , thereby avoiding the situation where the algorithm breaks down.

Since  $f(x)$  and  $b(x, \mu)$  have different characteristics, it makes sense to model them separately. In this case, we still have to indicate how the model  $m_{k,j}^b(x_{k,j} + s)$  of  $b(x_{k,j} + s, \mu_k)$  is related to  $b$  itself. We will suppose that AM.1–AM.3 hold, that is, that  $m_{k,j}^b$  is twice-continuously differentiable in the trust region, coincides with  $b$  at its centre, and has exact first-order derivatives. Furthermore, we will assume the equivalent of AC.4c on the model.

**AM.4h** For each  $k$  and each  $\epsilon > 0$ , there exists a constant<sup>203</sup>  $\kappa_{\text{bbmh}}(\epsilon, \mu_k) \geq 1$  such

---

<sup>203</sup>“bbmh” stands for “bound on the barrier model Hessian”.

that, for all  $k, j \geq 0$ ,

$$\|\nabla_{xx} m_{k,j}^b(x, \mu_k)\| \leq \kappa_{\text{bbmh}}(\epsilon, \mu_k)$$

for all  $x \in \mathcal{B}_{k,j} \cap \mathcal{C}$  such that  $\text{dist}(x, \partial\mathcal{C}) \geq \epsilon$ .

As before, we could simply assume in what follows that the model of the objective function,  $m_{k,j}^f$ , satisfies AM.1–AM.4. However, it is useful to note that we will only need AM.4 to hold on the relative interior of  $\mathcal{C}$ , since we will combine it with AM.4h to obtain the boundedness of the Hessian of the complete model  $m_{k,j}^f + m_{k,j}^b$ . This will allow us to consider the minimization of objective functions that have discontinuous second derivatives at the boundary of  $\mathcal{C}$ . Because of this possibility, we shall reformulate AM.4 in an appropriate way.

**AM.4i** The Hessian of the model remains bounded within the intersection of the trust region with the relative interior of the feasible region; that is,

$$\|\nabla_{xx} m_{k,j}^f(x)\| \leq \kappa_{\text{umh}} - 1$$

for all  $k, j \geq 0$  and for all  $x \in \mathcal{B}_{k,j} \cap \text{ri}\{\mathcal{C}\}$ .

We are now ready to specify our first primal<sup>204</sup> trust-region algorithm for the inner minimization problem.

**Algorithm 13.2.1: First primal inner algorithm**

**Step 0: Initialization.** An initial point  $x_{k,0} \in \text{ri}\{\mathcal{C}\}$  and an initial trust-region radius  $\Delta_{k,0}$  are given. The constants  $\eta_1, \eta_2, \gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Finally,  $\varsigma_k \in (0, 1)$  is also given. Compute  $\phi(x_{k,0}, \mu_k)$  and set  $j = 0$ .

**Step 1: Model definition.** Define a model  $m_{k,j}$  of  $\phi(x_{k,j} + s, \mu_k)$  in  $\mathcal{B}_{k,j}$  that is of the form

$$m_{k,j}(x_{k,j} + s) = m_{k,j}^f(x_{k,j} + s) + m_{k,j}^b(x_{k,j} + s),$$

where  $m_{k,j}^f$  is a model of  $f$  satisfying AM.1–AM.3 and AM.4i, and where  $m_{k,j}^b$  is a model of the barrier term  $b$  satisfying AM.1–AM.3 and AM.4h.

**Step 2: Step calculation.** Compute a step  $s_{k,j}$  that sufficiently reduces the model  $m_{k,j}$  in the sense of AA.1/AA.2 and such that  $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$ .

**Step 3: Acceptance of the trial point.** If  $x_{k,j} + s_{k,j} \notin \mathcal{C}$  or if

$$\text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) < \varsigma_k \text{dist}(x_{k,j}, \partial\mathcal{C}), \quad (13.2.1)$$

---

<sup>204</sup>Dual variables do not play any role in this algorithm.

set  $\rho_{k,j} = -\infty$ ,  $x_{k,j+1} = x_{k,j}$ , and go to Step 4; otherwise compute  $\phi(x_{k,j} + s_{k,j}, \mu_k)$  and define the ratio

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})}.$$

Then if  $\rho_{k,j} \geq \eta_1$ , define  $x_{k,j+1} = x_{k,j} + s_{k,j}$ ; otherwise define  $x_{k,j+1} = x_{k,j}$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Increment  $j$  by 1 and go to Step 1.

The only differences between this algorithm and Algorithm BTR, beside the fact that the objective function is now  $\phi(x, \mu_k)$  instead of  $f(x)$ , are the requirement that the initial point must lie in the interior of  $\mathcal{C}$  and the additional test in Step 3, which implies that iteration  $j$  is viewed as unsuccessful and  $\Delta_{k,j}$  is reduced whenever  $x_{k,j} + s_{k,j}$  falls in a region where the function  $b(x, \mu_k)$  is undefined. This obviously presumes that testing the inclusion  $x_{k,j} + s_{k,j} \in \text{ri}\{\mathcal{C}\}$  is computationally possible. Also note that the possibility of choosing  $\Delta_{k,j+1}$  as large as one wishes on very successful iterations is very important in practice, because it allows the trust-region radius to return to a reasonable value as soon as a successful step is made, instead of being constrained to remain of the order of magnitude of the distance from  $x_{k,j}$  to the boundary of  $\mathcal{C}$ . We have intentionally not specified how the parameter  $\varsigma_k$  is chosen for each inner minimization. This parameter specifies the minimum relative distance to the boundary that is acceptable in the course of the current minimization (see (13.2.1)). The fact that it is not fixed but may itself tend to zero makes fast asymptotic convergence of the outer iterates possible, but we do not discuss this issue in detail.

Are all the desirable convergence properties of Algorithm BTR still true for our first primal inner algorithm? Fortunately yes, as we now show.

We first consider the convergence to first-order critical points, as we did in Section 6.4. In order to apply the results of this section, we start by noticing that the iterates generated by Algorithm 13.2.1 will never become arbitrarily close to the boundary of  $\mathcal{C}$ . This crucial property is stated in the following simple proposition.

**Theorem 13.2.1** Suppose that AC1.c, AC.4c, and AC.8 hold and that  $\{x_{k,j}\}$  is a sequence of iterates generated by Algorithm 13.2.1 for  $k$  given, where  $m_{k,j}^f$  and  $m_{k,j}^b$  satisfy AC.2b, AM.1–AM.3, AM.4h, and AM.4i. Then there exists a constant<sup>205</sup>  $\kappa_{\text{mdb}}(k) \in (0, 1)$  such that

<sup>205</sup>“mdb” stands for “minimum distance to the boundary”.

$$\text{dist}(x_{k,j}, \partial\mathcal{C}) \geq \kappa_{\text{mdb}}(k)$$

for all  $j$ . Moreover, for all  $j$  and all  $x$  such that  $\|x - x_{k,j}\| \leq (1 - \varsigma_k) \text{dist}(x_j, \partial\mathcal{C})$ , we have that

$$\|\nabla_{xx} b(x, \mu)\| \leq \kappa_{\text{bbh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k) \quad (13.2.2)$$

and

$$\|\nabla_{xx} m_{k,j}^b(x_{k,j}, \mu)\| \leq \kappa_{\text{bbmh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k).$$

**Proof.** By AC.8, the level set  $\{x \in \mathcal{C} \mid b(x, \mu_k) \leq b(x_{k,0}, \mu_k)\}$ , and thus the level set of  $\phi(x_{k,0}, \mu_k)$ , must be bounded away from  $\partial\mathcal{C}$ . The existence of  $\kappa_{\text{mdb}}(k)$  then results from the inequality  $\phi(x_{k,j}, \mu_k) \leq \phi(x_{k,0}, \mu_k)$  which is true for all  $j \geq 0$ . Moreover, it can always be chosen small enough to ensure that it belongs to  $(0, 1)$ . The second part of the theorem follows from the first and from AC.4c and AM.4h.  $\square$

This result is crucial because it states that all arguments that use a sequence of trust-region radii  $\Delta_{k,j}$  converging to zero will be unhindered by the need to remain in the interior of  $\mathcal{C}$ . We now rephrase Theorem 6.4.1 (p. 133) on the error between predicted and exact objective values at the trial point as follows.

**Theorem 13.2.2** Suppose that AC.2b, AF.1, AF.3, AC.4c, AM.1–AM.3, AM.4h, and AM.4i hold. Then, if

$$\Delta_{k,j} \leq (1 - \varsigma_k) \kappa_{\text{mdb}}(k), \quad (13.2.3)$$

we have that

$$|\phi(x_{k,j} + s_{k,j}, \mu_k) - m_{k,j}(x_{k,j} + s_{k,j})| \leq \kappa_{\text{ubh}}(k) \Delta_{k,j}^2,$$

where

$$\kappa_{\text{ubh}}(k) \stackrel{\text{def}}{=} \max[\kappa_{\text{ufh}} + \kappa_{\text{bbh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k), \kappa_{\text{umh}} + \kappa_{\text{bbmh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k)].$$

**Proof.** The proof is identical to that of Theorem 6.4.1, except that we may only use AF.3 for  $f$ . We note that

$$\|s_{k,j}\| \leq \Delta_{k,j} \leq (1 - \varsigma_k) \kappa_{\text{mdb}}(k)$$

because of (13.2.3), and therefore, using (13.1.5),

$$\text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) \geq \text{dist}(x_{k,j}, \partial\mathcal{C}) - \|s_{k,j}\| \geq \varsigma_k \kappa_{\text{mdb}}(k) > 0,$$

where we have used Theorem 13.2.1 to deduce the last inequality.<sup>206</sup> Since  $\mathcal{C}$  is convex, we obtain that, for all  $\xi_{k,j} \in [x_{k,j}, x_{k,j} + s_{k,j}]$ ,

$$\text{dist}(\xi_{k,j}, \partial\mathcal{C}) \geq \min[\text{dist}(x_{k,j}, \partial\mathcal{C}), \text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C})] \geq \varsigma_k \kappa_{\text{mdb}}(k),$$

and thus, using (13.2.2) and AF.3, that

$$\|\nabla_{xx}\phi(\xi_{k,j}, \mu)\| \leq \kappa_{\text{ufh}} + \kappa_{\text{bbh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k).$$

Similarly,

$$\|\nabla_{xx}m_{k,j}(\xi_{k,j}, \mu_k)\| \leq \kappa_{\text{umh}} + \kappa_{\text{bbmh}}(\varsigma_k \kappa_{\text{mdb}}(k), \mu_k).$$

The proof is then concluded as for Theorem 6.4.1, keeping track of the modified constants.  $\square$

We may also restate Theorem 6.4.2 (p. 134) on very successful iterations in the same spirit.

**Theorem 13.2.3** Suppose that AC.2b, AF.1, AF.3, AC.1c, AC.4c, AC.8, AM.1–AM.3, AM.4h, AM.4i, and AA.1 hold. Suppose furthermore that  $g_{k,j} \neq 0$  and that

$$\Delta_{k,j} \leq \min \left[ \frac{\kappa_{\text{mdc}} \|g_{k,j}\| (1 - \eta_2)}{\kappa_{\text{ubh}}(k)}, (1 - \varsigma_k) \kappa_{\text{mdb}}(k) \right].$$

Then inner iteration  $j$  is very successful and

$$\Delta_{k,j+1} \geq \Delta_{k,j}.$$

**Proof.** The proof is identical to that of Theorem 6.4.2, except that the strengthened condition on  $\Delta_{k,j}$  is required for Theorem 13.2.2 to be true.  $\square$

If we suppose that AC.1c, AC.4c, AC.8, and AM.4h hold, we may now continue the convergence theory for the first primal inner-minimization algorithm in parallel with that of Algorithm BTR. Indeed, Theorem 6.4.3 (p. 135) remains true as stated, the only modification in its proof being that we now have to redefine

$$\kappa_{\text{lbd}}(k) \stackrel{\text{def}}{=} \min \left[ \frac{\gamma_1 \kappa_{\text{mdc}}(k) \kappa_{\text{lbg}}(1 - \eta_2)}{\kappa_{\text{ubh}}(k)}, (1 - \varsigma_k) \kappa_{\text{mdb}}(k) \right].$$

Theorems 6.4.4–6.4.6, 6.5.2, 6.5.5, 6.6.1–6.6.5, 6.6.7, and 6.6.8 (p. 136ff) also remain true if AF.2 and AM.5 are rephrased, when they appear, in terms of the functions  $f(x)$  and  $b(x, \mu)$ , as follows.

<sup>206</sup>Observe that we could replace (13.2.1) by an inequality of the form

$$\text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) < \min[\bar{\varsigma}, \varsigma_k \text{dist}(x_{k,j}, \partial\mathcal{C})]$$

and still obtain essentially the same result. The constant on the right-hand side would then be replaced by  $\min[\bar{\varsigma}, \varsigma_k \kappa_{\text{mdb}}(k)]$ .

**AW.2**  $\phi(x, \mu_k)$  is bounded below on  $\mathcal{C}$ ; that is, there exists a constant  $\kappa_{\text{lbb}}$  such that

$$\phi(x, \mu_k) \geq \kappa_{\text{lbb}}$$

for all  $k$  and all  $x \in \mathcal{C}$  and for some constant<sup>207</sup>  $\kappa_{\text{lbb}}$ .

This is a relatively strong assumption, since the barrier term itself does not satisfy it, but it nevertheless holds for many practical cases.

**AM.5c** We have that, for all  $k$ ,

$$\lim_{j \rightarrow \infty} \|\nabla_{xx} f(x_{k,j}) - \nabla_{xx} m_{k,j}^f(x_{k,j})\| = 0$$

and

$$\lim_{j \rightarrow \infty} \|\nabla_{xx} b(x_{k,j}, \mu_k) - \nabla_{xx} m_{k,j}^b(x_{k,j})\| = 0$$

whenever

$$\lim_{j \rightarrow \infty} \|\nabla_x m_{k,j}^f(x_{k,j}) + \nabla_x m_{k,j}^b(x_{k,j}, \mu_k)\| = 0.$$

Observe that this assumption differs from AM.5 in that the Hessians of the objective function and its model must asymptotically coincide when a first-order critical point of  $\phi(x, \mu)$  (and not of  $f(x)$ ) is approached. AM.6 also needs to be completed for the barrier term.

**AC.6b** We have that, for all  $k, j$ ,

$$\|\nabla_{xx} m_{k,j}^b(x) - \nabla_{xx} m_{k,j}^b(y)\| \leq \kappa_{\text{ich}} \|x - y\|$$

for all  $x, y \in \mathcal{B}_{k,j}$ .

Here, we might possibly have increased the value of  $\kappa_{\text{ich}}$  to ensure both this bound and the original AM.6.

As a consequence of AM.5c and AC.6b, we obtain that AM.5 and AM.6 hold for the barrier function  $\phi(x, \mu)$ . We may thus deduce that *the convergence properties of Algorithm 13.2.1 are identical to those of Algorithm BTR*. Hence minimizing a barrier function with a trust-region method is, at least in theory, only marginally more difficult than purely unconstrained minimization.

## Notes and References for Section 13.2

The form of (13.2.1) has many variants. A very general one is the condition

$$\text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) < \psi_k(\text{dist}(x_{k,j}, \partial\mathcal{C})),$$

where  $\psi_k$  is a forcing function such that  $\psi_k(d) < d$  for all  $d > 0$ . Our development corresponds to the choice  $\psi_k(d) = \varsigma_k d$ .

---

<sup>207</sup>“lbb” stands for “lower bound on the barrier”.

### 13.3 Constrained Cauchy and Eigenpoints

From a practical point of view, the decision to deem an iteration, for which  $x_{k,j} + s_{k,j}$  lies outside  $\mathcal{C}$ , unsuccessful is somewhat inefficient, since we might need a number of these pointless iterations before finally ensuring that the trial point falls within the feasible region, each of which requires the solution of a trust-region subproblem. Even if the subproblem is solved only approximately, the computational overhead may be substantial and, as we shall see in this section, often unnecessary.

One possibility is to make these unsuccessful iterations implicit, by which we mean that we would not compute an (approximate) step if there is ever a risk of it resulting in an infeasible trial point, a most extreme case being when the trust region intersects the infeasible domain. However, even if we do not actually compute the step, for our theory to apply, we still have to exhibit a value of  $s_k$  giving sufficient model decrease at each of these implicit iterations for which the associated trial point satisfies (13.2.1). A suitable  $s_k$  is far from obvious, and thus this approach remains theoretically unsound. Furthermore, even if we could prove global convergence, the steps resulting from such a strategy would necessarily be shorter than the distance from the current iterate to the boundary of the feasible domain. The resulting algorithm might then be forced to take many successive tiny steps to make progress along this boundary.

We therefore investigate an alternative approach. In what follows, we consider keeping the trust-region radius as it is—thus allowing the trust region to extend into the infeasible region—but, at the same time, imposing the further restriction that the step must be feasible, by explicitly requiring that (13.2.1) must not hold. The price to pay for adding this additional constraint may be small. For instance, if  $\mathcal{C}$  is the positive orthant, and “distance” and the trust region are defined in the  $\ell_\infty$  norm, these resulting conditions on the step still specify a “box-shaped” region, which is similar in shape to the trust-region  $\mathcal{B}_{k,j}$  itself. In other cases, for instance when an  $\ell_2$  norm is used with the above  $\mathcal{C}$ , the subproblem is defined over the intersection of a sphere and a box, which is a cumbersome geometry to work with. In particular, the very efficient algorithms for solving the  $\ell_2$ -norm problem we encountered in Chapter 7 are no longer appropriate. Thus, although the method we are about to define has some strong advantages, it most certainly does not supersede our first primal inner algorithm in all cases.

We specify the resulting second primal inner algorithm as follows.

**Algorithm 13.3.1: Second primal inner barrier algorithm**

**Step 0: Initialization.** An initial point  $x_{k,0} \in \text{ri}\{\mathcal{C}\}$  and an initial trust-region radius  $\Delta_{k,0}$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Finally, a  $\varsigma_k \in (0, 1)$  is also given. Compute  $\phi(x_{k,0}, \mu_k)$  and set  $j = 0$ .

**Step 1: Model definition.** Define a model  $m_{k,j}$  of  $\phi(x_{k,j} + s, \mu_k)$  in  $\mathcal{B}_{k,j}$ , which is of the form

$$m_{k,j}(x_{k,j} + s) = m_{k,j}^f(x_{k,j} + s) + m_{k,j}^b(x_{k,j} + s),$$

where  $m_{k,j}^f$  is a model of  $f$  and where  $m_{k,j}^b$  is a model of the barrier term  $b$ , both satisfying AM.1–AM.3, AM.4h, and AM.4i.

**Step 2: Step calculation.** Define  $d_{k,j} = \text{dist}(x_{k,j}, \partial\mathcal{C})$ . Compute a step  $s_{k,j}$  such that

$$x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j} \cap \mathcal{C} \quad \text{and} \quad \text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k d_{k,j} \quad (13.3.1)$$

and such that it sufficiently reduces the model  $m_{k,j}$  in the sense that

$$\begin{aligned} & m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j}) \\ & \geq \kappa_{\text{sod}} \max \left\{ \left\| g_{k,j} \right\| \min \left[ \frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, (1 - \varsigma_k)d_{k,j} \right], \right. \\ & \quad \left. - \tau_{k,j} \min \left[ \tau_{k,j}^2, \Delta_{k,j}^2, (1 - \varsigma_k)^2 d_{k,j}^2 \right] \right\} \end{aligned} \quad (13.3.2)$$

**Step 3: Acceptance of the trial point.** Compute  $\phi(x_{k,j} + s_{k,j}, \mu_k)$  and define the ratio

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})}.$$

Then if  $\rho_{k,j} \geq \eta_1$ , define  $x_{k,j+1} = x_{k,j} + s_{k,j}$ ; otherwise define  $x_{k,j+1} = x_{k,j}$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

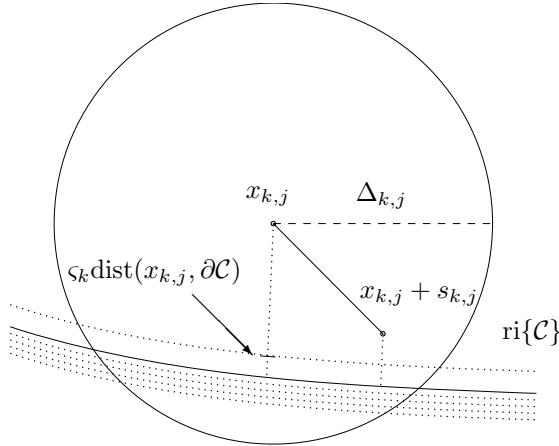
Increment  $j$  by 1 and go to Step 1.

Figure 13.3.1 on the next page shows an example of an iteration of this algorithm at which the step  $s_{k,j}$  satisfies the constraints (13.3.1)<sup>208</sup> while the trust region extends into the infeasible domain.

Does this new strategy for choosing the step  $s_{k,j}$  ensure global convergence of the algorithm? We can of course answer positively, if we can prove that it implies that AA.1/AA.1b and AA.2/AA.2b hold. This is the path we shall follow for the rest of this section.

---

<sup>208</sup>Which we illustrate for a more general  $\mathcal{C}$  than the positive orthant.

Figure 13.3.1: An acceptable step  $s_{k,j}$  in Algorithm 13.3.1.

As we have seen more than once by now, the first step is to show that it is possible to find a value that satisfies (13.3.2); the point we shall exhibit is a generalization of the Cauchy point. The technique is very simple: as we now require that the second part of condition (13.3.1) holds, we simply impose this same condition on our new Cauchy point, which we therefore call a constrained Cauchy point. It is then easy to verify that this property is satisfied if we redefine the Cauchy arc to be

$$x_{k,j}^{\text{CC}}(t) \stackrel{\text{def}}{=} \{x \mid x = x_{k,j} - tg_{k,j}, t \geq 0, t\|g_{k,j}\| \leq (1 - \varsigma_k)d_{k,j} \text{ and } x \in \mathcal{B}_k\}, \quad (13.3.3)$$

that is, if we restrict the Cauchy arc to the line segment starting at the current iterate, moving in the direction of the negative gradient and stopping before we get too close to the boundary of  $\mathcal{C}$  (or on the trust-region boundary if this happens first). For simplicity, we only consider the case where the model is quadratic of the form

$$m_{k,j}(x_{k,j} + s) = m_{k,j}(x_{k,j}) + \langle g_{k,j}, s \rangle + \frac{1}{2}\langle s, H_{k,j}s \rangle. \quad (13.3.4)$$

The *constrained Cauchy point*

$$x_{k,j}^{\text{CC}} = x_{k,j} - t_{k,j}^{\text{CC}}g_{k,j} \quad (13.3.5)$$

is now defined as the solution of the problem

$$\min_{t \geq 0} m_{k,j}(x_{k,j} - tg_{k,j}),$$

such that

$$x_{k,j} - tg_{k,j} \in \mathcal{B}_{k,j} \text{ and } t\|g_{k,j}\| \leq (1 - \varsigma_k)d_{k,j}. \quad (13.3.6)$$

In fact, this definition is more restrictive than what would be possible. While our discussion of the step indicates that we would be satisfied with any step for which the trial point is not too close to the boundary, the second part of (13.3.3) requires that the modified Cauchy point lies in a ball surrounding the current iterate, whose radius is such

that no point in the ball is too close to the boundary.<sup>209</sup> Figure 13.3.2 illustrates this, where the dark-shaded area indicates the ball in which we have restricted our search for the constrained Cauchy point, while the light-shaded area indicates that in which a trial point is allowed to lie. The relative differences between the two areas indicated in Figure 13.3.2 may be significantly reduced (or increased) if a single norm is used.

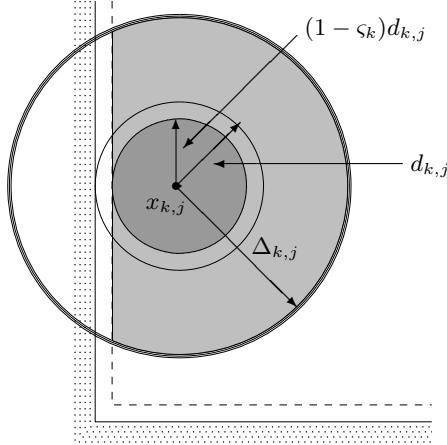


Figure 13.3.2: The areas for acceptable constrained Cauchy points (dark-shaded) and acceptable trial steps (light-shaded).

We may then prove the following on the model reduction associated with the constrained Cauchy point.

**Theorem 13.3.1** If the model is of the form (13.3.4) and if we define the Cauchy point by (13.3.5)–(13.3.6), we have that

$$m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{\text{CC}}) \geq \frac{1}{2} \|g_{k,j}\| \min \left[ \frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, (1 - s_k)d_{k,j} \right]. \quad (13.3.7)$$

**Proof.** This proof parallels that of Theorem 6.3.1 (p. 125). We first note that, for all  $t \geq 0$ ,

$$m_{k,j}(x_{k,j} - tg_{k,j}) = m_{k,j}(x_{k,j}) - t\|g_{k,j}\|^2 + \frac{1}{2}t^2 \langle g_{k,j}, H_{k,j}g_{k,j} \rangle. \quad (13.3.8)$$

We now consider the case where the curvature of the model along the steepest-descent direction is positive, that is, when

$$\langle g_{k,j}, H_{k,j}g_{k,j} \rangle > 0, \quad (13.3.9)$$

and compute the value of the parameter  $t$  at which the unique minimum of (13.3.8) is attained. Let us denote this optimal parameter by  $t_{k,j}^*$ . Differentiating (13.3.8)

<sup>209</sup>Such situations were already discussed at the end of Section 6.3.4.

with respect to  $t$  and equating the result to zero, we obtain that

$$0 = \|g_{k,j}\|^2 - t_{k,j}^* \langle g_{k,j}, H_{k,j} g_{k,j} \rangle,$$

which immediately gives that

$$t_{k,j}^* = \frac{\|g_{k,j}\|^2}{\langle g_{k,j}, H_{k,j} g_{k,j} \rangle}. \quad (13.3.10)$$

Two subcases may then occur. The first is when this minimizer satisfies both parts of (13.3.6), that is, when

$$t_{k,j}^* \|g_{k,j}\| \leq \min[\Delta_{k,j}, (1 - \varsigma_k) d_{k,j}].$$

Then  $t_{k,j}^{CC} = t_{k,j}^*$  and we may replace this expression in the model decrease (13.3.8), which allows us to deduce that

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^C) &\geq \frac{\|g_{k,j}\|^4}{\langle g_{k,j}, H_{k,j} g_{k,j} \rangle} - \frac{1}{2} \frac{\|g_{k,j}\|^4}{\langle g_{k,j}, H_{k,j} g_{k,j} \rangle} \\ &\geq \frac{\|g_{k,j}\|^2}{2\beta_{k,j}}, \end{aligned} \quad (13.3.11)$$

where we used the fact that  $|\langle g_{k,j}, H_{k,j} g_{k,j} \rangle| \leq \|g_{k,j}\|^2 \beta_{k,j}$  because of the Cauchy–Schwarz inequality and the definition of  $\beta_{k,j}$ . If, on the other hand,

$$t_{k,j}^* \|g_{k,j}\| > \min[\Delta_{k,j}, (1 - \varsigma_k) d_{k,j}], \quad (13.3.12)$$

then the line minimum is outside the trust region or not sufficiently feasible, and we have that

$$t_{k,j}^{CC} \|g_{k,j}\| = \min[\Delta_{k,j}, (1 - \varsigma_k) d_{k,j}]. \quad (13.3.13)$$

Combining (13.3.10), (13.3.12), and (13.3.13), we see that

$$\langle g_{k,j}, H_{k,j} g_{k,j} \rangle \leq \frac{\|g_{k,j}\|^2}{t_{k,j}^{CC}}.$$

Substituting this last inequality in (13.3.8) and using (13.3.13), we obtain that

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{CC}) &= t_{k,j}^{CC} \|g_{k,j}\|^2 - \frac{1}{2} [t_{k,j}^{CC}]^2 \langle g_{k,j}, H_{k,j} g_{k,j} \rangle \\ &\geq t_{k,j}^{CC} \|g_{k,j}\|^2 - \frac{1}{2} t_{k,j}^{CC} \|g_{k,j}\|^2 \\ &= \frac{1}{2} \|g_{k,j}\| \min[\Delta_{k,j}, (1 - \varsigma_k) d_{k,j}]. \end{aligned} \quad (13.3.14)$$

Finally, we consider the case where the curvature of the model along the steepest-descent direction is negative, that is, when (13.3.9) is violated. We then obtain from (13.3.8) that

$$m_{k,j}(x_{k,j} - tg_{k,j}) = m_{k,j}(x_{k,j}) - t\|g_{k,j}\|^2 + \frac{1}{2} t^2 \langle g_{k,j}, H_{k,j} g_{k,j} \rangle \leq m_{k,j}(x_{k,j}) - t\|g_{k,j}\|^2 \quad (13.3.15)$$

for all  $t \geq 0$ . In that case, it is easy to see that (13.3.13) holds. Combining this equality and (13.3.15), we deduce that

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{\text{CC}}) &\geq \|g_{k,j}\| \min[\Delta_{k,j}, (1 - \varsigma_k)d_{k,j}] \\ &\geq \frac{1}{2}\|g_{k,j}\| \min[\Delta_{k,j}, (1 - \varsigma_k)d_{k,j}]. \end{aligned} \quad (13.3.16)$$

We may then conclude our proof by noting that (13.3.11), (13.3.14), and (13.3.16) imply that (13.3.7) holds.  $\square$

A similar result holds for the case where the model is not quadratic, but we do not give details of the proof. The result of this theorem is essentially identical to the form of the sufficient model decrease assumption, AA.1, except that a third term has now appeared in the minimum. However, this is not serious, as we show in the following easy proposition.

**Corollary 13.3.2** Suppose that AC.1c, AC.2b, AC.4c, and AC.8 hold and that  $\{x_j\}$  is a sequence of iterates generated by Algorithm 13.3.1, where  $m_j^f$  and  $m_j^b$  satisfy AM.1–AM.3, AM.4i, and AM.4h. If the model is of the form (13.3.4) and if we define the Cauchy point by (13.3.5)–(13.3.6), then there are constants  $\kappa_{\text{mdc}} > 0$  and  $\kappa_{\text{mdb}} > 0$ , both depending only on  $k$ , such that

$$m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{\text{CC}}) \geq \kappa_{\text{mdc}} \pi_{k,j} \min \left[ \frac{\pi_{k,j}}{\beta_{k,j}}, \Delta_{k,j} \right], \quad (13.3.17)$$

where

$$\pi_{k,j} = \min[\|g_{k,j}\|, (1 - \varsigma_k)\kappa_{\text{mdb}}]. \quad (13.3.18)$$

**Proof.** We first observe that our assumption allows us to apply Theorem 13.2.1 for the second primal inner algorithm and thus to deduce that there exists  $\kappa_{\text{mdb}} > 0$  such that  $\text{dist}(x_j, \partial\mathcal{C}) \geq \kappa_{\text{mdb}}$ . We then obtain that

$$(1 - \varsigma_k)d_{k,j} \geq (1 - \varsigma_k)\kappa_{\text{mdb}} \stackrel{\text{def}}{=} \kappa_0. \quad (13.3.19)$$

Inserting this bound in (13.3.7), we obtain that

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j}^{\text{CC}}) &\geq \frac{1}{2}\|g_{k,j}\| \min \left[ \frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, \kappa_0 \right] \\ &\geq \frac{1}{2}\|g_{k,j}\| \min \left[ \frac{\min[\|g_{k,j}\|, \kappa_0]}{\beta_{k,j}}, \Delta_{k,j} \right] \\ &\geq \frac{1}{2}\pi_{k,j} \min \left[ \frac{\pi_{k,j}}{\beta_{k,j}}, \Delta_{k,j} \right], \end{aligned}$$

where we have used the inequalities  $\beta_{k,j} \geq 1$  and the definition of  $\pi_{k,j}$ . This yields the desired conclusion with  $\kappa_{\text{mdc}} = \frac{1}{2}$ .  $\square$

The attentive reader will have already noticed that (13.3.18) is a first-order criticality measure for the inner minimization problem and that (13.3.17) is the corresponding form of AA.1b. As a consequence, we may conclude that, for each  $k$ ,

$$\lim_{j \rightarrow \infty} \pi_{k,j} = 0,$$

and therefore that *every limit point of the sequence of iterates produced by the second primal inner algorithm is first-order critical*, provided the assumptions of Corollary 13.3.2 hold. Moreover, the convergence results of Section 6.5 also apply to this algorithm. Similar statements hold when the model is not quadratic.

Precisely the same technique is used to prove convergence to second-order critical points: we first generalize the notion of eigenpoint to that of a *constrained eigenpoint* by requiring that the step to the eigenpoint satisfies both parts of (13.3.1), which then allows us to obtain AA.2b and the desired convergence results follow.

The notion of eigenpoint is generalized as follows. In the constrained context, let us assume, for simplicity of exposition, that the model is a quadratic of the form (13.3.4). If  $H_{k,j} = \nabla_{xx} m_{k,j}(x_{k,j})$  is indefinite and  $\tau_{k,j} < 0$  is its leftmost eigenvalue, we then determine a direction  $u_{k,j}$  such that

$$\langle u_{k,j}, g_{k,j} \rangle \leq 0, \quad \|u_{k,j}\| = \min[\Delta_{k,j}, (1 - \varsigma_k)d_{k,j}] \quad (13.3.20)$$

and

$$\langle u_{k,j}, H_{k,j} u_{k,j} \rangle \leq \kappa_{\text{snc}} \tau_{k,j} \min[\Delta_{k,j}, (1 - \varsigma_k)d_{k,j}]^2$$

for some constant  $\kappa_{\text{snc}} \in (0, 1]$ . Then the constrained eigenpoint, which we denote by  $x_{k,j}^{\text{CE}}$ , is, not surprisingly, defined at the minimum of the model on all points of the form  $x_{k,j} + t u_{k,j}$  for  $t \in [0, 1]$ . A simple adaptation of the proof of Theorem 6.6.1 (p. 149) (where we replace  $\Delta_k$  by  $\min[\Delta_{k,j}, (1 - \varsigma_k)d_{k,j}]$ ) then yields the following bound on the model decrease.

**Theorem 13.3.3** Suppose that the assumptions of Corollary 13.3.2 hold, together with AM.5b, and that  $H_k$  has a negative eigenvalue  $\tau_k$ . Then we have that

$$m_k(x_k) - m_k(x_k^{\text{CE}}) \geq -\frac{1}{2} \kappa_{\text{snc}} \tau_k \min[\Delta_k, (1 - \varsigma_k)d_{k,j}]^2.$$

This, together with (13.3.19), allows us to conclude that AA.2b holds (with  $\kappa_{\text{lsd}} = (1 - \varsigma_k)\kappa_{\text{mdb}}$ ), and therefore, according to our discussion of Section 8.1.3, that *the second-order convergence results of Section 6.6 hold for the second primal inner algorithm* (Algorithm 13.3.1). As one might expect, the same conclusion holds if we allow nonquadratic models.

## Notes and References for Section 13.3

Byrd (1999) discusses possible approximate methods to solve the constrained step calculation, including backtracking and specialized dogleg strategies.

## 13.4 The Primal Log-Barrier Algorithm

We now return to the solution of the constrained problem

$$\min_{x \geq 0} f(x) \quad (13.4.1)$$

using a barrier function approach. We first analyse the case where we use the log-barrier function (13.1.1). As we suggested at the beginning of this chapter, we proceed by (approximately) solving a sequence of problems of the form (13.1.6), where the barrier parameter  $\mu$  goes to zero. Each of these minimizations is carried out using the first or the second primal inner algorithm. Before describing this procedure in detail, we introduce an additional piece of notation. Given an iterate  $x > 0$ , we define

$$X = \text{diag}([x]_1, \dots, [x]_n),$$

the diagonal matrix whose  $(i, i)$ th entry is  $[x]_i$ . We can now formally state our constrained minimization technique, which we call the primal barrier algorithm.

### Algorithm 13.4.1: Primal barrier algorithm

**Step 0: Initialization.** An initial point  $x_0 > 0$  and an initial barrier parameter  $\mu_0 > 0$  are given. The forcing functions (see Section 3.3.3)  $\epsilon^D(\mu)$  and  $\epsilon^E(\mu)$  are also given. Set  $k = 0$ .

**Step 1: Inner minimization.** Choose a value  $\varsigma_k \in (0, 1)$ . Approximately minimize the log-barrier function  $\phi^{\log}(x, \mu_k) = f(x) - \mu_k \langle e, \log(x) \rangle$  starting from  $x_k$  and using Algorithm 13.2.1 or 13.3.1, in which

$$m_{k,j}^b(x_{k,j} + s) = \mu_k \left( -\langle e, \log(x_{k,j}) \rangle - \langle X_{k,j}^{-1}e, s \rangle + \frac{1}{2} \langle s, X_{k,j}^{-2}s \rangle \right). \quad (13.4.2)$$

Stop this algorithm as soon as an iterate  $x_{k,j} = x_{k+1}$  is found for which

$$\|\nabla_x f(x_{k+1}) - \mu_k X_{k+1}^{-1}e\| \leq \epsilon^D(\mu_k), \quad (13.4.3)$$

$$\lambda_{\min}[\nabla_{xx} f(x_{k+1}) + \mu_k X_{k+1}^{-2}] \geq -\epsilon^E(\mu_k), \quad (13.4.4)$$

and

$$x_{k+1} > 0. \quad (13.4.5)$$

**Step 2: Update the barrier parameter.** Choose  $\mu_{k+1} > 0$  in such a way as to ensure that

$$\lim_{k \rightarrow \infty} \mu_k = 0. \quad (13.4.6)$$

Increment  $k$  by 1 and return to Step 1.

This description calls for several comments. The first is that we have not specified in which way the sequence  $\{\mu_k\}$  has to converge to zero. This leaves much freedom in Step 2 and does not require, for instance, that the sequence of barrier parameters monotonically decreases with  $k$ . The second observation is that the two “tolerance functions”  $\epsilon^D(\mu)$  and  $\epsilon^E(\mu)$  are not specified in detail either, except that they have to be forcing functions, which implies that, for all  $k$ ,

$$\epsilon^D(\mu_k) > 0 \text{ and } \epsilon^E(\mu_k) > 0 \quad (13.4.7)$$

since  $\mu_k > 0$  for all  $k$ . The third is that we still have to verify that applying one of our two primal inner algorithms to the problem

$$\min_{x \geq 0} \phi^{\log}(x, \mu_k) = f(x) - \mu_k \langle e, \log(x) \rangle$$

produces an iterate satisfying conditions (13.4.3)–(13.4.5) after a finite number of iterations, making Step 1 well defined. Finally, and most importantly, we aim to prove that, if the sequence  $\{x_k\}$  is well defined, it converges to a second-order critical point. We start by showing that Step 1 is well defined.

**Lemma 13.4.1** The log-barrier term (13.1.2) satisfies AC.1c, AC.4c, and AC.8. Moreover, AM.1–AM.3, AM.4h, AM.5c, and AC.6b also hold for its model (13.4.2).

**Proof.** The elementary properties of the logarithm imply that AC.1c and AC.8 hold. Furthermore, the identity

$$\nabla_{xx} b^{\log}(x, \mu) = \mu X^{-2} \quad (13.4.8)$$

then ensures that AC.4c and AM.4h are also satisfied. Finally, the fact that  $m_{k,j}^b$  is a quadratic second-order model of the barrier term (13.4.2) implies that AM.1–AM.3, AM.5c, and AC.6b hold.  $\square$

Note that we have not proved that AM.5c holds in its entirety, but merely that its part related to the model of the barrier term does. This is why we still have to assume that the part related to the objective function holds in the convergence analysis that follows.

**Theorem 13.4.2** Suppose that AF.1 and AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose finally that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$ . Then a point  $x_{k+1}$  satisfying (13.4.3)–(13.4.5) is obtained after a finite number of iterations of Algorithm 13.2.1 or 13.3.1.

**Proof.** Because of Lemma 13.4.1, we see that AM.6 for  $m_{k,j}^f$  and AC.6b ensure AM.6 for  $m_{k,j}$ . Furthermore, AM.5c is nothing other than AM.5 for this model and the statement of the primal inner algorithms implies that sufficient model reduction of the form specified by AA.1 and AA.2 is obtained. Finally, AW.2 corresponds to AF.2 for the problem of minimizing the log-barrier function. Thus, we may apply the conclusion of Section 13.2 and deduce, from Theorem 6.4.6 (p. 137), that

$$\lim_{j \rightarrow \infty} \|\nabla_x \phi^{\log}(x_{k,j}, \mu_k)\| = \lim_{j \rightarrow \infty} \|\nabla_x f(x_{k,j}) - \mu_k X_{k,j}^{-1} e\| = 0.$$

We also obtain, from Theorem 6.6.4 (p. 154) and (13.4.8), that

$$\limsup_{j \rightarrow \infty} \lambda_{\min}[\nabla_{xx} \phi^{\log}(x_{k,j}, \mu_k)] = \limsup_{j \rightarrow \infty} \lambda_{\min}[\nabla_{xx} f(x_{k,j}) + \mu_k X_{k,j}^{-2}] \geq 0.$$

These limits together with (13.4.7) then guarantee that we can define  $x_{k+1} = x_{k,j}$  for any  $j$  sufficiently large to ensure that (13.4.3) and (13.4.4) are satisfied. Furthermore, since both primal inner algorithms maintain strict feasibility of the iterates, we also obtain that (13.4.5) holds.  $\square$

Thus the sequence  $\{x_k\}$  is in turn well defined, and we now show that weak second-order necessary conditions are asymptotically obtained. To be more precise, we first introduce the notion of a *consistently active subsequence* of iterates. We say that a subsequence  $\{x_{k_j}\}$  of iterates is consistently active with respect to the constraints  $x \geq 0$  if, for each  $i = 1, \dots, n$ , either

$$\lim_{j \rightarrow \infty} [x_{k_j}]_i = 0 \text{ or } \liminf_{j \rightarrow \infty} [x_{k_j}]_i > 0.$$

This is to say that each bound constraint is asymptotically active or inactive for the complete subsequence. We also define the set of asymptotically active constraints for such a subsequence by

$$\mathcal{A}\{x_{k_j}\} \stackrel{\text{def}}{=} \{i \in \{1, \dots, n\} \mid \lim_{j \rightarrow \infty} [x_{k_j}]_i = 0\}.$$

In other words, the set of asymptotically active bounds is fixed for the iterates of a consistently active subsequence. Since there are only a finite number of such sets, as each constraint is asymptotically active or is not, the number of consistently active subsequences is finite for any sequence  $\{x_k\}$  of nonnegative iterates. Furthermore, the complete sequence of iterates may be partitioned into disjoint consistently active subsequences. Observe also that, if  $\{x_k\}$  has limit points, then each subsequence converging to a specific limit point  $x_*$  is consistently active, as the set of asymptotically active bounds is then determined by the components of  $x_*$ , that is

$$\mathcal{A}\{x_{k_j}\} = \{i \in \{1, \dots, n\} \mid [x_*]_i = 0\}.$$

We then have the following result.

**Theorem 13.4.3** Suppose that AF.1–AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose finally that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$  and that  $\{x_k\}$  is a sequence of iterates generated by the primal barrier algorithm. Then, we have that

$$\liminf_{k \rightarrow \infty} [\nabla_x f(x_k)]_i \geq 0 \quad (i = 1, \dots, n). \quad (13.4.9)$$

Furthermore, we also have that, for every consistently active subsequence of iterates  $\{x_{k_\ell}\}$ ,

$$\lim_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0 \quad (i \notin \mathcal{A}\{x_{k_\ell}\}) \quad (13.4.10)$$

and

$$\liminf_{\ell \rightarrow \infty} \langle u, [\nabla_{xx} f(x_{k_\ell})]u \rangle \geq 0 \quad (13.4.11)$$

for every vector  $u$  for which  $[u]_i = 0$  whenever  $i \in \mathcal{A}\{x_{k_\ell}\}$ .

**Proof.** Suppose that  $\{x_{k_\ell}\}$  is a consistently active subsequence and

$$\mathcal{A} = \mathcal{A}\{x_{k_\ell}\} \text{ and } \mathcal{F} = \{1, \dots, n\} \setminus \mathcal{A},$$

the index sets corresponding to the asymptotically active and inactive (free) variables, respectively. If  $i \in \mathcal{F}$ , the limit (13.4.6), the forcing nature of  $\epsilon^D$ , and (13.4.3) ensure that

$$\lim_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0.$$

If  $i \in \mathcal{A}$ , two cases may occur. Either

$$\liminf_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0$$

or this limit inferior is nonzero. In the second case, we obtain from (13.4.6), (13.4.3), and the definition of  $\epsilon^D(\mu)$  that

$$\liminf_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = \liminf_{\ell \rightarrow \infty} \frac{\mu_{k_\ell-1}}{[x_{k_\ell}]_i} \geq 0,$$

where the last inequality follows from the positivity of  $[x_{k_\ell}]_i$  and  $\mu_{k_\ell-1}$  for all  $j$ , which itself results from the definition of the barrier parameter and (13.4.5). Thus (13.4.9) and (13.4.10) are satisfied.

Suppose now that (13.4.11) does not hold, that is, that there exists a unit vector  $u$  and a subsequence  $\{x_{k_{\ell_t}}\} \subseteq \{x_{k_\ell}\}$  such that

$$[u]_i = 0 \text{ if } i \in \mathcal{A} \text{ and } \liminf_{t \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_{\ell_t}})u \rangle = -\epsilon \quad (13.4.12)$$

for some  $\epsilon > 0$ . But, since (13.4.6) implies that

$$\lim_{t \rightarrow \infty} \mu_{k_{\ell_t}-1} X_{k_{\ell_t}}^{-2} e_i = 0$$

for  $i \notin \mathcal{A}$ , the first part of (13.4.12) then ensures that

$$\lim_{t \rightarrow \infty} \mu_{k_{\ell_t}-1} \langle u, X_{k_{\ell_t}}^{-2} u \rangle = 0.$$

Hence we deduce from this limit that

$$\liminf_{t \rightarrow \infty} \langle u, [\nabla_{xx} f(x_{k_{\ell_t}}) + \mu_{k_{\ell_t}-1} X_{k_{\ell_t}}^{-2}] u \rangle = \liminf_{t \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_{\ell_t}}) u \rangle = -\epsilon.$$

But this contradicts (13.4.4) for  $j$  sufficiently large. Hence no vector satisfying (13.4.12) can exist, (13.4.11) holds, and the theorem is proved.  $\square$

When there are finite limit points, this theorem immediately implies the following result.

**Corollary 13.4.4** Suppose that AF.1–AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose finally that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$  and that  $x_*$  is a limit point of the sequence of iterates generated by the primal barrier Algorithm 13.4.1. Then  $x_*$  is a weak second-order critical point.

**Proof.** It is enough to take  $\{x_{k_j}\}$  converging to  $x_*$  in Theorem 13.4.3.  $\square$

However, while Theorem 13.4.2 ensures that the inner iteration always terminates, it does not ensure that there are finite limit points. In fact, nothing prevents each sequence of inner iterates from diverging to plus infinity,<sup>210</sup> in which case the sequence of outer iterates may also diverge. Theorem 13.4.3 then merely shows that weak second-order necessary conditions asymptotically hold along consistently active subsequences, which is all we may reasonably ask for.

The reader may wonder, at this point, if it is possible to ensure that limit points of the sequence  $\{x_k\}$  are not only weak second-order critical points, but strong ones. Unfortunately, this cannot be guaranteed in general, as is shown by the following simple example. Consider a bound-constrained quadratic program of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x^T H x, \\ x \in \mathbb{R}^n & \\ x \geq 0 & \end{array} \quad (13.4.13)$$

where  $H$  is the symmetric, indefinite  $n \times n$  matrix defined by

$$H = I - \frac{3}{2} \frac{zz^T}{\|z\|^2} \quad (13.4.14)$$

with  $z = e - ne_1$ . It is easy to see that

$$z^T e = e^T e - ne_1^T e = n - n = 0, \quad (13.4.15)$$

$$z^T e_1 = e^T e_1 - ne_1^T e_1 = 1 - n, \quad (13.4.16)$$

---

<sup>210</sup>This would happen, for instance, if one minimized  $e^{-x}$  on all  $x \geq 0$ .

and

$$\|z\|_2^2 = e^T e + n^2 e_1^T e_1 - 2n e_1^T e = n + n^2 - 2n = n(n-1). \quad (13.4.17)$$

The definitions (13.4.14) and (13.4.15) together imply that

$$He = e. \quad (13.4.18)$$

Now choose a sequence  $\{\mu_k\}$  of barrier parameters converging to zero and let

$$x_{k+1} = \mu_k^{\frac{1}{2}} e. \quad (13.4.19)$$

We now show that  $x_{k+1}$  is a minimizer of the problem

$$\underset{\substack{x \in \mathbb{R}^n \\ x \geq 0}}{\text{minimize}} \quad \phi^{\log}(x, \mu_k)$$

that satisfies second-order *sufficient* optimality conditions for this problem. Indeed, the first-order optimality condition holds since

$$\nabla_x \phi^{\log}(x_{k+1}, \mu_k) = Hx_{k+1} - \mu_k X_{k+1}^{-1} e = \mu_k^{\frac{1}{2}} e - \mu_k^{1-\frac{1}{2}} e = \mu_k^{\frac{1}{2}}(e - e) = 0,$$

where we used (13.4.18) and (13.4.19), and thus (13.4.3) holds. Moreover, we have that

$$\nabla_{xx} \phi^{\log}(x_{k+1}, \mu_k) = H + \mu_k X_{k+1}^{-2} = \frac{1}{2}I + \frac{3}{2} \left( I - \frac{zz^T}{\|z\|_2^2} \right)$$

is obviously positive definite since the first term of the last right-hand side is positive definite and the last term in brackets is an orthogonal projector, which is therefore positive semidefinite. Thus (13.4.4) also holds. As expected,  $\{x_{k+1}\}$  converges to  $x_* = 0$ , the only critical point of problem (13.4.13). However, using the fact that all constraints are linear as well as (13.4.14), (13.4.16), and (13.4.17), we find that

$$e_1^T \nabla_{xx} \ell(x, y) e_1 = e_1^T H e_1 = 1 - \frac{3}{2} \frac{(e_1^T z)^2}{\|z\|_2^2} = 1 - \frac{3(n-1)^2}{2n(n-1)} = \frac{3-n}{2n},$$

which is strictly negative for all values of  $n \geq 4$ . But

$$e_1 \in \mathcal{N}_+ = \{x \in \mathbb{R}^n \mid x \geq 0\},$$

where  $\mathcal{N}_+$  is defined as in (3.2.12) (p. 41), and thus the strong second-order necessary conditions do not hold at  $x_*$ . We conclude from this counterexample that we cannot expect the strong second-order necessary conditions to hold in general, when the sequence of iterates is generated by the primal barrier algorithm.

## Notes and References for Section 13.4

The notion of consistently active subsequences originates in the use of “fickle” variables by Chvátal (1983), p. 37, in the context of linear programming: these are variables whose status never settles down to being asymptotically active or asymptotically inactive. The term

“faithful” was also used by Gould (1991) to denote consistently active subsequences in the context of quadratic programming.

The counterexample showing that strong second-order necessary conditions cannot be expected to hold in general is extracted from Gould and Toint (1999), where the same result is shown for a wider class of barrier functions, including the reciprocal barrier functions discussed in the next paragraph.

## 13.5 Reciprocal Barriers

The logarithmic barrier term (13.1.2) is not the only possible suitable one, and we note here that one could also consider *reciprocal barrier terms* defined, for the nonnegativity constraints on the variables, by

$$b^{R(\alpha)}(x, \mu) = \mu \sum_{i=1}^n \frac{1}{\alpha[x]_i^\alpha},$$

where  $\alpha \geq 1$  is an additional parameter. They correspond to a stronger singularity, and a resulting stronger repulsive effect, of the barrier function on the boundary of the feasible region than occurs with the log-barrier function, the strength of the singularity increasing with  $\alpha$ . This is illustrated in Figure 13.5.1 on the next page, where the levels curves of the logarithmic and reciprocal barrier function ( $\mu = 2$ ) for the problem (13.1.4) are shown for increasing values of  $\alpha$ . The corresponding increasing distance of the minimizer from the boundary and stronger effect of the singularity (as materialized by the density of the level curves along  $\partial\mathcal{C}$ ) are clearly apparent.

It is then easy to define a reciprocal primal barrier algorithm along the lines of the primal barrier algorithm, but where (13.4.2) is then replaced by

$$m_{k,j}^b(x_{k,j} + s) = \mu_k \left( \sum_{i=1}^n \frac{1}{\alpha[x_{k,j}]_i^\alpha} - \langle X_{k,j}^{-(\alpha+1)} e, s \rangle + \frac{1}{2}(\alpha+1)\langle s, X_{k,j}^{-(\alpha+2)} s \rangle \right).$$

Our analysis applies straightforwardly to this algorithm, since it is easy to verify that a variant of Lemma 13.4.1 also holds in this case. However, reciprocal barriers do not seem to be used very much in practice, and we will not consider them any further.

## Notes and References for Section 13.5

The reciprocal barrier was introduced by Carroll (1959) in his Ph.D. thesis. The idea was, at the time, supported by encouraging numerical results but was not theoretically justified. Further practical and theoretical work on this idea includes Carroll (1961); Fiacco and McCormick (1963), where convergence was proved for convex problems; Fiacco and McCormick (1964a, 1964b); and El-Bakry (1998).

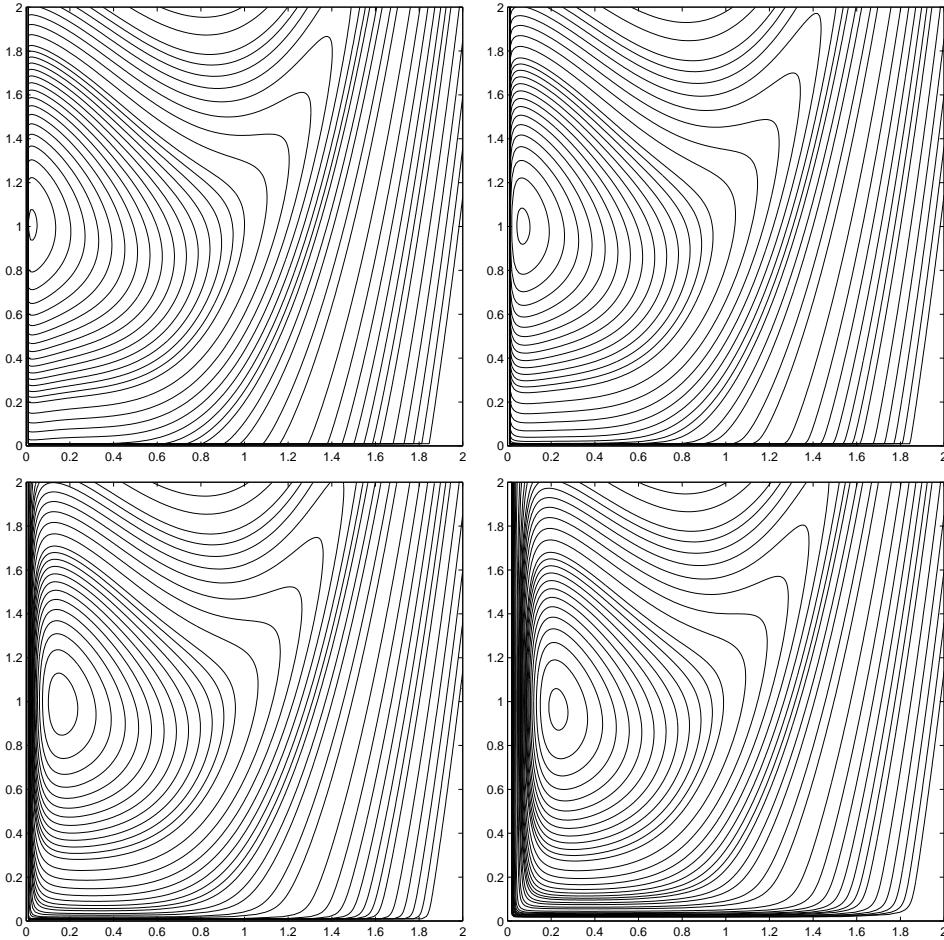


Figure 13.5.1: The level curves of barrier functions for problem (13.1.4) using different barrier terms for  $\mu = 2$ : the logarithmic barrier function is used in the top left picture,  $R(\frac{1}{2})$  in the top right,  $R(1)$  in the bottom left, and  $R(2)$  in the bottom right.

## 13.6 A Primal-Dual Algorithm

### 13.6.1 The Algorithm

When applying primal barrier algorithms to problem (13.4.1) in practice, it is most apparent that convergence of the inner iterates  $x_{k,j}$  slows down considerably whenever they happen to be close to the boundary of the feasible set. This is because the singularity of the logarithm then dominates, which means that quadratic models of the log-barrier function—while adequate very locally—do not fit the barrier function well, as the Hessian matrix,  $\nabla_{xx}f(x_{k,j}) + \mu_k X_{k,j}^{-2}$ , of the log-barrier function for (13.4.1) is dominated by its second, highly nonquadratic term. As a consequence, if a variable becomes small, the change in this variable resulting from the “ideal” Newton step is of the order of the value of the variable itself, which is small. Progress away from this bound, when necessary, is thus very slow.

One possible way of alleviating this numerical problem is to abandon the analytic expression for the local second-order behaviour of the barrier term ( $\mu_k X_{k,j}^{-2}$ ), and to replace it by a term whose growth would be, one hopes, less dominant. In primal-dual methods, one chooses to replace

$$\nabla_{xx} m_{k,j}(x_{k,j}) + \mu_k X_{k,j}^{-2} \quad \text{by} \quad \nabla_{xx} m_{k,j}(x_{k,j}) + X_{k,j}^{-1} Z_{k,j},$$

where  $Z_{k,j}$  is a bounded positive diagonal matrix.

Interestingly, there is another way to motivate this modification of the Hessian of the model, using a perturbation argument. Consider the first-order necessary conditions for the problem of minimizing a model  $m(x)$  of the objective function on the feasible region, namely,

$$\nabla_x m(x) - z = 0, \quad XZ = 0, \quad x \geq 0, \quad z \geq 0, \quad (13.6.1)$$

where  $z$  is the vector of Lagrange multipliers or dual variables<sup>211</sup> and where  $Z = \text{diag}([z]_1, \dots, [z]_n)$ . The second equation of (13.6.1) is known as the *complementarity condition* for the problem. Notice that it expresses a true combinatorial requirement: if a variable is nonzero, then its corresponding dual variable must be zero, and vice-versa. As combinatorial conditions may be very hard to verify, especially in large dimensions, we perturb the complementarity condition to avoid this combinatorial aspect. Introducing  $\mu > 0$ , a small perturbation parameter, we then write

$$\nabla_x m(x) - z = 0, \quad XZ = \mu e, \quad x \geq 0, \quad z \geq 0.$$

Newton's equation for this system of nonlinear equations at the iterate  $(x_{k,j}, z_{k,j})$  are

$$\begin{aligned} \nabla_{xx} m_{k,j}(x_{k,j}) \Delta x_{k,j} - \Delta z_{k,j} &= -g_{k,j} + z_{k,j}, \\ X_{k,j} \Delta z_{k,j} + Z_{k,j} \Delta x_{k,j} &= \mu_k e - X_{k,j} Z_{k,j} e, \\ x_{k,j} + \Delta x_{k,j} &\geq 0, \quad z_{k,j} + \Delta z_{k,j} \geq 0. \end{aligned} \quad (13.6.2)$$

Substituting the second equation into the first, we obtain

$$\left[ \nabla_{xx} m_{k,j}(x_{k,j}) + X_{k,j}^{-1} Z_{k,j} \right] \Delta x_{k,j} = -\left[ g_{k,j} - \mu_k X_{k,j}^{-1} e \right].$$

We then note that the right-hand side of this relation is nothing but the negative gradient of the log-barrier function,  $-\nabla_x \phi^{\log}(x_{k,j}, \mu_k)$ . Hence  $\Delta x_{k,j}$  may be interpreted as a quasi-Newton step for this barrier function, where the Hessian of the model has been modified by replacing  $\mu_k X_{k,j}^{-2}$  by  $X_{k,j}^{-1} Z_{k,j}$ . This is exactly what we proposed above, except that we may now interpret  $z_{k,j}$  as the vector of dual variables. If the barrier parameter  $\mu_k$  is fixed, the resulting primal-dual<sup>212</sup> inner algorithm<sup>213</sup> is Algorithm 13.6.1.

---

<sup>211</sup>In this chapter, we use the notation  $z$  instead of  $y$  for multipliers associated with inequality constraints in order to conform to a well-established tradition in the analysis of primal-dual methods.

<sup>212</sup>Since it now involves the vector of dual variables.

<sup>213</sup>Strictly speaking, we should call it the *second* primal-dual inner algorithm, as it extends the *second* primal inner algorithm. However, although we could develop a primal-dual version of the first primal inner algorithm in exactly the same way (this would then be our first primal-dual inner algorithm), we choose to ignore this subtlety in what follows for the sake of brevity.

**Algorithm 13.6.1: Primal-dual inner algorithm**

**Step 0: Initialization.** An initial point  $x_{k,0} \in \text{ri}\{\mathcal{C}\}$ , a vector of dual variables  $z_{k,0} > 0$ , and an initial trust-region radius  $\Delta_{k,0}$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Finally, the constant  $\varsigma_k \in (0, 1)$  is also given. Compute  $f(x_{k,0})$  (if not already known) and set  $j = 0$ .

**Step 1: Model definition.** Define a model  $m_{k,j}$  of  $\phi^{\log}(x_{k,j} + s, \mu_k)$ , which is of the form

$$m_{k,j}(x_{k,j} + s) = m_{k,j}^f(x_{k,j} + s) \quad (13.6.3)$$

$$- \mu_k \left[ \langle e, \log(x_{k,j}) \rangle + \langle X_{k,j}^{-1} e, s \rangle \right] - \frac{1}{2} \langle s, X_{k,j}^{-1} Z_{k,j} s \rangle$$

in  $\mathcal{B}_{k,j}$ , where  $m_{k,j}^f$  is a model of  $f$  satisfying AM.1–AM.3 and AM.4i.

**Step 2: Step calculation.** Define  $d_{k,j} = \text{dist}(x_{k,j}, \partial\mathcal{C})$ . Compute a step  $s_{k,j}$  such that  $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$  and  $\text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k d_{k,j}$ , and such that it sufficiently reduces the model  $m_{k,j}$  in the sense that

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j}) \\ \geq \kappa_{\text{sod}} \max \left\{ \left\| g_{k,j} \right\| \min \left[ \frac{\|g_{k,j}\|}{\beta_{k,j}}, \Delta_{k,j}, (1 - \varsigma_k)d_{k,j} \right], \right. \\ \left. - \tau_{k,j} \min \left[ \tau_{k,j}^2, \Delta_{k,j}^2, (1 - \varsigma_k)^2 d_{k,j}^2 \right] \right\}. \end{aligned} \quad (13.6.4)$$

**Step 3: Acceptance of the trial point.** Compute  $\phi^{\log}(x_{k,j} + s_{k,j}, \mu_k)$  and

$$\rho_{k,j} = \frac{\phi^{\log}(x_{k,j}, \mu_k) - \phi^{\log}(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})}.$$

Then if  $\rho_{k,j} \geq \eta_1$ , define  $x_{k,j+1} = x_{k,j} + s_{k,j}$ ; otherwise define  $x_{k,j+1} = x_{k,j}$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

**Step 5: Update the dual variables.** Define  $z_{k,j+1} > 0$ . Increment  $j$  by 1 and go to Step 1.

At first sight, the modification to the second primal inner algorithm appears to be minimal, requiring only the introduction of the dual variable  $z_{k,j}$ . Furthermore, if the choice

$$z_{k,j} = \mu_k X_{k,j}^{-1} e$$

is made, then both algorithms are identical. However, considerable numerical experience has shown that this modification may be crucial in terms of numerical efficiency,<sup>214</sup> at least for some choices of the dual variables. Note that the exact choice of  $z_{k,j}$  has not been specified in the primal-dual inner algorithm beyond the stipulation that it must be strictly positive.

Of course, we need to insert this algorithm for inner minimization in an outer iteration that drives the barrier/perturbation parameter to zero. Again, this algorithm is very similar to the primal barrier algorithm and only differs from its predecessor in that the stopping criterion for the inner iterations now involves the values of the dual variables.

**Algorithm 13.6.2: Primal-dual barrier algorithm**

**Step 0: Initialization.** An initial point  $x_0 > 0$ , a vector of initial dual variables  $z_0 > 0$  and an initial barrier parameter  $\mu_0 > 0$  are given. The forcing functions  $\epsilon^D(\mu)$ ,  $\epsilon^E(\mu)$ , and  $\epsilon^C(\mu)$  are also given. Set  $k = 0$ .

**Step 1: Inner minimization.** Choose a value  $s_k \in (0, 1)$ . Approximately minimize the log-barrier function  $\phi^{\log}(x, \mu_k) = f(x) - \mu_k \langle e, \log(x) \rangle$  using Algorithm 13.6.1. Stop this algorithm as soon as an iterate  $(x_{k,j}, z_{k,j}) = (x_{k+1}, z_{k+1})$  is found for which

$$\|\nabla_x f(x_{k+1}) - z_{k+1}\| \leq \epsilon^D(\mu_k), \quad (13.6.5)$$

$$\|X_{k+1} Z_{k+1} - \mu_k I\| \leq \epsilon^C(\mu_k), \quad (13.6.6)$$

$$\lambda_{\min}[\nabla_{xx} f(x_{k+1}) + X_{k+1}^{-1} Z_{k+1}] \geq -\epsilon^E(\mu_k) \quad (13.6.7)$$

and

$$x_{k+1} > 0 \text{ and } z_{k+1} > 0. \quad (13.6.8)$$

**Step 2: Update the barrier parameter.** Choose  $\mu_{k+1} > 0$  in such a way to ensure that

$$\lim_{k \rightarrow \infty} \mu_k = 0. \quad (13.6.9)$$

Increment  $k$  by 1 and return to Step 1.

---

<sup>214</sup>For the reason explained at the beginning of this subsection.

### 13.6.2 Convergence Properties

Once again, we start by examining conditions under which the inner algorithm converges to first- and second-order critical points of the log-barrier function  $\phi^{\log}(x, \mu_k)$ . Fortunately, the developments of Section 13.2 provide a useful basis, as we merely have to verify that the barrier term satisfies AC.1c, AC.4c, and AC.8, which we have already done in the first part of Lemma 13.4.1, and we also need to establish conditions under which the new model for the barrier term

$$m_{k,j}^b(x_{k,j} + s, \mu) = -\mu_k \left[ \langle e, \log(x_{k,j}) \rangle + \langle X_{k,j}^{-1}e, s \rangle \right] + \frac{1}{2} \langle s, X_{k,j}^{-1}Z_{k,j}s \rangle$$

satisfies AM.1–AM.3, AM.4h, AM.5c, and AC.6b. That  $m_{k,j}^b$  satisfies AM.1–AM.3 and AC.6b immediately follows from its definition. Thus only the boundedness of the Hessian of the model (AM.4h) and its convergence to the Hessian of the barrier term near first-order critical points (AM.5c) remain in question. These are ensured by the following assumptions, AA.7 and AA.8, respectively.

**AA.7** For each  $k \geq 0$ , there exists a constant<sup>215</sup>  $\kappa_{\text{uzi}} > 0$  such that

$$[z_{k,j}]_i \leq \kappa_{\text{uzi}} \max \left[ \frac{1}{[x_{k,j}]_i}, 1 \right]$$

for all  $j \geq 0$  and all  $i = 1, \dots, n$ .

Note that AA.7 immediately implies that there exists a constant<sup>216</sup>  $\kappa_{\text{ubz}} > 0$  such that

$$\|z_{k,j}\| \leq \kappa_{\text{ubz}} \max \left[ \|X_{k,j}^{-1}e\|, 1 \right] \quad (13.6.10)$$

for all  $j \geq 0$ .

**AA.8** For each  $k \geq 0$ ,

$$\lim_{j \rightarrow \infty} \|z_{k,j} - \mu_k X_{k,j}^{-1}e\| = 0 \quad \text{if} \quad \lim_{j \rightarrow \infty} \|g_{k,j}\| = 0.$$

We then immediately obtain the desired result.

**Lemma 13.6.1** The log-barrier term (13.1.2) satisfies AC.1c, AC.4c, and AC.8. If AA.7 and AA.8 also hold, then AM.1–AM.3, AM.4h, AM.5c, and AC.6b are satisfied for its model (13.6.3).

**Proof.** The proof of the first part is identical to that used in Lemma 13.4.1. We have just seen that AM.1–AM.3 and AC.6b result from the definition of the model (13.6.3). Finally, since

$$\nabla_{xx} m_{k,j}^b(x_{k,j} + s) = X_{k,j}^{-1}Z_{k,j},$$

<sup>215</sup>“uzi” stands for “upper bound on  $z_i$ ”.

<sup>216</sup>“ubz” stands for “upper bound on  $z$ ”.

we see that AM.4h follows from (13.6.10) and also that AM.5c follows from AA.8 and (13.4.8).  $\square$

As was the case for the primal barrier algorithm, this lemma ensures that all the conditions for the asymptotic second-order convergence of the inner algorithm are again satisfied.

**Theorem 13.6.2** Suppose that AF.1 and AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose finally that AA.7 and AA.8 hold, as well as AW.2 for  $\phi^{\log}(x, \mu_k)$ . Then a point  $x_{k+1}$  satisfying (13.6.5)–(13.6.8) is obtained after a finite number of iterations of Algorithm 13.6.1.

**Proof.** The proof is identical to that of Theorem 13.4.2, except that we have to use the fact that

$$\lim_{j \rightarrow \infty} \|\nabla_x m_{k,j}(x_{k,j})\| = \lim_{j \rightarrow \infty} \|g_{k,j}\| = 0$$

and AA.8 to obtain that

$$\lim_{j \rightarrow \infty} \|z_{k,j} - \mu_k X_{k,j}^{-1} e\| = 0.$$

We are then able to verify that (13.6.5)–(13.6.7) hold for all  $j$  sufficiently large because of Theorems 6.4.6 (p. 137) and 6.6.4 (p. 154).  $\square$

We now obtain the following result on the asymptotic second-order convergence of the primal-dual algorithm.

**Theorem 13.6.3** Suppose that AF.1–AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose finally that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$ , and that AA.7 and AA.8 hold. If  $\{x_k\}$  is a sequence of iterates generated by Algorithm 13.6.2, then we have that

$$\liminf_{k \rightarrow \infty} [\nabla_x f(x_k)]_i \geq 0 \quad (i = 1, \dots, n). \quad (13.6.11)$$

Furthermore, we also obtain that, for every consistently active subsequence of iterates  $\{x_{k_\ell}\}$ ,

$$\lim_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0 \quad (i \notin \mathcal{A}\{x_{k_\ell}\}) \quad (13.6.12)$$

and

$$\liminf_{\ell \rightarrow \infty} \langle u, [\nabla_{xx} f(x_{k_\ell})] u \rangle \geq 0 \quad (13.6.13)$$

for each vector  $u$  such that  $[u]_i = 0$  whenever  $i \in \mathcal{A}\{x_{k_\ell}\}$ .

**Proof.** The proof parallels that of Theorem 13.4.3. As in the proof of Theorem 13.6.2, we obtain from the convergence of the inner iteration to a first-order critical point and AA.8 that

$$\lim_{j \rightarrow \infty} \|z_{k,j} - \mu_k X_{k,j}^{-1} e\| = 0. \quad (13.6.14)$$

Suppose now that  $\{x_{k_\ell}\}$  is a consistently active subsequence and that

$$\mathcal{A} = \mathcal{A}\{x_{k_\ell}\} \text{ and } \mathcal{F} = \{1, \dots, n\} \setminus \mathcal{A}$$

are the index sets corresponding to the asymptotically active and inactive (free) variables, respectively. If  $i \in \mathcal{F}$ , the limit (13.6.9), the forcing nature of  $\epsilon^D$ , (13.6.5), and (13.6.14) ensure that

$$\lim_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0.$$

If  $i \in \mathcal{A}$ , two cases may occur. Either

$$\liminf_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = 0$$

or this limit inferior is nonzero. In that case, we obtain from (13.6.9), (13.6.5), (13.6.14), and the definition of  $\epsilon^D(\mu)$  that

$$\liminf_{\ell \rightarrow \infty} [\nabla_x f(x_{k_\ell})]_i = \liminf_{\ell \rightarrow \infty} \frac{\mu_{k_\ell-1}}{[x_{k_\ell}]_i} \geq 0,$$

where the last inequality follows from the positivity of  $[x_{k_\ell}]_i$  and  $\mu_{k_\ell-1}$  for all  $\ell$ , which itself results from the definition of the barrier parameter and (13.6.8). Thus (13.6.11) and (13.6.12) are satisfied.

Suppose now that (13.6.13) does not hold, that is, that there exists a unit vector  $u$  and a subsequence  $\{x_{k_{\ell_t}}\} \subseteq \{x_{k_\ell}\}$  such that

$$[u]_i = 0 \text{ if } i \in \mathcal{A} \text{ and } \liminf_{t \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_{\ell_t}}) u \rangle = -\epsilon \quad (13.6.15)$$

for some  $\epsilon > 0$ . But, since (13.6.9) and (13.6.14) imply that

$$\lim_{t \rightarrow \infty} X_{k_{\ell_t}}^{-1} Z_{k_{\ell_t}} e_i = 0$$

for  $i \notin \mathcal{A}$ , the first part of (13.6.15) then ensures that

$$\lim_{t \rightarrow \infty} \langle u, X_{k_{\ell_t}}^{-1} Z_{k_{\ell_t}} u \rangle = 0.$$

Hence we deduce, from this limit, that

$$\liminf_{t \rightarrow \infty} \langle u, [\nabla_{xx} f(x_{k_{\ell_t}}) + X_{k_{\ell_t}}^{-1} Z_{k_{\ell_t}}] u \rangle = \liminf_{t \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_{\ell_t}}) u \rangle = -\epsilon.$$

But this contradicts (13.6.7) for  $\ell$  sufficiently large. Hence no vector satisfying (13.6.15) can exist, and (13.6.13) holds.  $\square$

Finally, the conclusions of Corollary 13.4.4 remain true for Algorithm 13.6.2 so long as AA.7 and AA.8 also hold. Thus the convergence properties of the primal-dual algorithm are as good as may reasonably be expected.

### 13.6.3 Updating the Vector of Dual Variables

We conclude this section on the primal-dual barrier method by indicating how the dual variables  $z_{k,j+1}$  may be updated in practice at Step 5 of the primal-dual inner algorithm, Algorithm 13.6.1, while ensuring AA.7 and AA.8. A simple idea is to use the value predicted in the middle part of the Newton equations (13.6.2), which is

$$\bar{z}_{k,j+1} = \mu_k X_{k,j}^{-1} e - X_{k,j}^{-1} Z_{k,j} s_{k,j}. \quad (13.6.16)$$

However, there is no guarantee that the choice  $z_{k,j+1} = \bar{z}_{k,j+1}$  maintains feasibility of the dual variables ( $z_{k,j+1} \geq 0$ ), nor that it satisfies AA.7 or AA.8. We thus need to safeguard it, which can be achieved by projecting (componentwise) the value (13.6.16) into the interval

$$\mathcal{I} = \left[ \kappa_{\text{zul}} \min \left( e, z_{k,j}, \mu_k X_{k,j+1}^{-1} e \right), \max \left( \kappa_{\text{zuu}} e, z_{k,j}, \kappa_{\text{zuu}} \mu_k^{-1} e, \kappa_{\text{zuu}} \mu_k X_{k,j+1}^{-1} e \right) \right], \quad (13.6.17)$$

where  $\kappa_{\text{zul}}$  and  $\kappa_{\text{zuu}}$  are constants<sup>217</sup> such that

$$0 < \kappa_{\text{zul}} < 1 < \kappa_{\text{zuu}}. \quad (13.6.18)$$

This is to say that

$$z_{k,j+1} = \begin{cases} P_{\mathcal{I}}[\bar{z}_{k,j+1}] & \text{if } x_{k,j+1} = x_{k,j} + s_{k,j}, \\ z_{k,j} & \text{if } x_{k,j+1} = x_{k,j}, \end{cases} \quad (13.6.19)$$

where  $P_{\mathcal{I}}[v]$  is the componentwise projection of the vector  $v$  onto the interval  $\mathcal{I}$ . Note that the top part of (13.6.19) corresponds to the case where iteration  $j$  is successful, that is,  $j \in \mathcal{S}$ , while the bottom part corresponds to the case where  $j \notin \mathcal{S}$ . Also note that the interval  $\mathcal{I}$  always contains the choices

$$z_{k,j+1} = z_{k,j} \text{ and } z_{k,j+1} = \mu_k X_{k,j+1}^{-1} e,$$

the latter corresponding to the pure primal method. Does this safeguarded value satisfy the required conditions? We now verify that this is usually the case.

**Theorem 13.6.4** Suppose that AF.1–AF.3 hold for the objective function  $f(x)$ . Suppose also that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6, and that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$ . Suppose finally that  $\{x_{k,j}, z_{k,j}\}$  is a sequence of primal and dual iterates generated, at a given outer iteration  $k$ , by Algorithm 13.6.2, where  $z_{k,j+1}$  is updated according to (13.6.19), with  $\mathcal{I}$  being given by (13.6.17) and  $\bar{z}_{k,j+1}$  by (13.6.16). Then  $z_{k,j+1} > 0$  and AA.7 holds. If, furthermore,

$$\lim_{j \rightarrow \infty} \|s_{k,j}\| = 0 \text{ when } \lim_{j \rightarrow \infty} \|g_{k,j}\| = 0, \quad (13.6.20)$$

then AA.8 is also satisfied.

---

<sup>217</sup>“zul” and “zuu” stand for “ $\underline{z}$ ’s update lower bound” and “ $\underline{z}$ ’s update upper bound”, respectively. In practice,  $\kappa_{\text{zul}} = \frac{1}{2}$  and  $\kappa_{\text{zuu}} = 10^{20}$  appear to work satisfactorily.

**Proof.** The positivity of the vector of dual variables immediately results from the fact that the lower end of the interval  $\mathcal{I}$  is always positive. To obtain its boundedness, we notice that our assumptions allow us to apply Theorem 13.2.1 and deduce that

$$[x_{k,j}]_i \geq \kappa_{\text{mdb}} \quad (13.6.21)$$

for all  $j \geq 0$  and all  $i \in \{1, \dots, n\}$ . The definition of  $\mathcal{I}$  and this bound then imply that, for each  $i = 1, \dots, n$ ,

$$[z_{k,j}]_i \leq \max \left[ \kappa_{\text{zuu}}, [z_{k,0}]_i, \frac{\kappa_{\text{zuu}}}{\mu_k}, \frac{\kappa_{\text{zuu}} \mu_k}{\kappa_{\text{mdb}}} \right] \stackrel{\text{def}}{=} \kappa_{\text{ubz}}$$

and AA.7 follows. Suppose now that the  $g_{k,j}$  converge to zero, which implies, because of (13.6.20), that

$$\lim_{j \rightarrow \infty} \|s_{k,j}\| = 0. \quad (13.6.22)$$

Then (13.6.21) ensures that

$$\lim_{\substack{j \rightarrow \infty \\ j \in \mathcal{S}}} \|X_{k,j}^{-1} - X_{k,j+1}^{-1}\| = \lim_{\substack{j \rightarrow \infty \\ j \in \mathcal{S}}} \max_{i=1,\dots,n} \left[ \frac{|[s_{k,j}]_i|}{|[x_{k,j}]_i [x_{k,j+1}]_i|} \right] \leq \lim_{j \rightarrow \infty} \frac{\|s_{k,j}\|}{\kappa_{\text{mdb}}^2} = 0. \quad (13.6.23)$$

But, since

$$\begin{aligned} \|\bar{z}_{k,j+1} - \mu_k X_{k,j+1}^{-1} e\| &\leq \|\bar{z}_{k,j+1} - \mu_k X_{k,j}^{-1} e\| + \mu_k \|(X_{k,j+1}^{-1} - X_{k,j+1}^{-1})e\| \\ &\leq \|X_{k,j}^{-1} Z_{k,j} e\| \|s_{k,j}\| + \mu_k \sqrt{n} \|X_{k,j+1}^{-1} - X_{k,j+1}^{-1}\|, \end{aligned}$$

where we have used (13.6.16), we also obtain from (13.6.21), (13.6.22), AA.7, and (13.6.23) that

$$\lim_{\substack{j \rightarrow \infty \\ j \in \mathcal{S}}} \|\bar{z}_{k,j+1} - \mu_k X_{k,j+1}^{-1} e\| = 0.$$

Now this limit and (13.6.18) give that, for  $j \in \mathcal{S}$  sufficiently large,

$$\kappa_{\text{zuu}} \mu_k X_{k,j+1}^{-1} e \leq \bar{z}_{k,j+1} \leq \kappa_{\text{zuu}} \mu_k X_{k,j+1}^{-1} e.$$

Hence, from the definition of  $z_{k,j+1}$ , we have that  $z_{k,j+1} = \bar{z}_{k,j+1}$  for  $j \in \mathcal{S}$  sufficiently large. Thus (13.6.16) yields that

$$X_{k,j+1} Z_{k,j+1} e = X_{k,j+1} X_{k,j}^{-1} (-Z_{k,j} s_{k,j} + \mu_k e). \quad (13.6.24)$$

On the other hand, since

$$\frac{[x_{k,j+1}]_i}{[x_{k,j}]_i} = \frac{[x_{k,j} + s_{k,j}]_i}{[x_{k,j}]_i} = 1 + \frac{[s_{k,j}]_i}{[x_{k,j}]_i},$$

we deduce from (13.6.21) and (13.6.22) that

$$\lim_{\substack{j \rightarrow \infty \\ j \in \mathcal{S}}} X_{k,j+1} X_{k,j}^{-1} = I.$$

We then obtain from this limit, AA.7, (13.6.21), and (13.6.24) that

$$\lim_{\substack{j \rightarrow \infty \\ j \in \mathcal{S}}} X_{k,j+1} Z_{k,j+1} e = \mu_k e.$$

AA.8 then follows because  $z_{k,j+1} = z_{k,j}$  for  $j \notin \mathcal{S}$ , that is, exactly when  $x_{k,j+1} = x_{k,j}$ .  $\square$

## 13.7 Scaling of Barrier Methods

We now return to the important question of scaling and trust-region geometry, an issue that we have not discussed thus far in the context of constrained problems. We saw in Section 6.7 and Chapter 7 how crucial this idea is for unconstrained problems, and it is therefore vital that we analyse how it applies to constrained ones.

### 13.7.1 Reintroducing Iteration-Dependent Norms

As we have seen in Section 6.7, scaling is best taken into account by defining the norm in the space of variables in a way that takes the underlying geometry of the problem into account. We also discussed, in Sections 8.1 and 8.1.4, how to translate crucial conditions like those on model decrease at the Cauchy or eigenpoints in a language that uses scaled quantities instead of unscaled ones. In particular, we now know that if displacements are measured in the norm  $\|\cdot\|_{k,j}$  at iteration  $(k,j)$ , then it is natural to measure the gradient of the function we are trying to minimize in the dual norm  $\|\cdot\|_{[k,j]}$  and the size of the involved Hessians in the bidual norm  $\|\cdot\|_{\{k,j\}}$  (see (8.1.13) [p. 260]). Thus we now return to our practice of choosing a specific norm  $\|\cdot\|_{k,j}$  at each iteration (at the beginning of Step 1 of inner-minimization algorithms). How should we rephrase some of our earlier results in this chapter to take this more general context into account? There are several issues to consider.

The first issue is that condition (13.2.1), which states that the trial point should not be too close to being infeasible, may now be expressed in the scaled norm. This is achieved by using this norm to redefine

$$\text{dist}_{k,j}(y, \mathcal{Z}) \stackrel{\text{def}}{=} \inf_{z \in \mathcal{Z}} \|y - z\|_{k,j}.$$

As a result, we also need to rephrase AC.4c, AC.8, and AM.4h by adapting the notion of (unscaled) distances  $\text{dist}(\cdot)$  to that of scaled distances  $\text{dist}_{k,j}(\cdot)$ , and also to introduce the bidual norms  $\|\cdot\|_{\{k,j\}}$  to measure the Hessians in the last two as well as in AM.4i. Another consequence is that Theorem 13.2.1 must also be expressed in scaled distance and (13.2.2) made to use the bidual norms.

A second important issue is that we need to restate our sufficient model decrease condition for the Cauchy point and, if necessary eigenpoint, so that the correct norms

are used. In this light, (13.3.2) becomes

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j}) \\ \geq \kappa_{\text{sod}} \max \left\{ \|g_{k,j}\|_{[k,j]} \min \left[ \frac{\|g_{k,j}\|_{[k,j]}}{\beta_{\{k,j\}}}, \Delta_{k,j}, (1 - \varsigma_k) d_{k,j}^s \right], \right. \\ \left. - \tau_{k,j} \min \left[ \tau_{k,j}^2, \Delta_{k,j}^2, (1 - \varsigma_k)^2 (d_{k,j}^s)^2 \right] \right\}, \end{aligned} \quad (13.7.1)$$

where  $d_{k,j}^s = \text{dist}_{k,j}(x_{k,j}, \partial\mathcal{C})$  is now defined in terms of the scaled norm  $\|\cdot\|_{k,j}$ . The same transformation applies to (13.6.4). The results concerning constrained Cauchy and eigenpoints remain essentially unmodified, modulo the above change in norms. For instance, (13.3.6) and (13.3.20) become, respectively,

$$x_{k,j} - t g_{k,j} \in \mathcal{B}_{k,j} \text{ and } t \|g_{k,j}\|_{k,j} \leq (1 - \varsigma_k) d_{k,j}^s$$

and

$$\langle u_{k,j}, g_{k,j} \rangle \leq 0, \quad \|u_{k,j}\|_{k,j} = \min[\Delta_{k,j}, (1 - \varsigma_k) d_{k,j}^s],$$

where  $d_{k,j}^s$  is expressed in the scaled norm.

The third point is to notice that the dual variables  $z_{k,j}$ , whose use is fundamental to the primal-dual methods, must be measured in the dual norm. This is immediately clear from, say, (13.6.1), given that we already noticed that the gradient is naturally measured in this norm. Hence AA.8 must now be expressed in the dual norm, (13.6.5) becomes

$$\|\nabla_x f(x_{k+1}) - z_{k+1}\|_{[k]} \leq \epsilon^D(\mu_k),$$

and (13.6.20) becomes

$$\lim_{j \rightarrow \infty} \|s_{k,j}\|_{k,j} = 0 \text{ when } \lim_{j \rightarrow \infty} \|g_{k,j}\|_{[k,j]} = 0.$$

Interestingly, (13.6.6) remains expressed in the  $\ell_2$  norm.

It is then crucial but easy to deduce that *all convergence properties of our barrier algorithms are preserved, provided one is ready to assume that all norms remain uniformly equivalent* (that is, that AN.1 holds).

### 13.7.2 Scaling of the Inner Iterations

If we suppose that the Hessian of the model for the log-barrier function for (13.4.1) is positive definite at a given inner iterate, it seems natural (see Section 7.7) to consider its Hessian as a local preconditioning metric, that is, to choose

$$\|\cdot\|_{k,j} = \|\cdot\|_{\nabla_{xx} m_{k,j}(x_{k,j})}.$$

However, as this metric includes, for the primal-dual method, the term  $X_{k,j}^{-1} Z_{k,j}$ , whose curvature tends to infinity as the inner iterates  $x_{k,j}$  approach  $\partial\mathcal{C}$ , the reader may

wonder if AN.1 still holds. Fortunately, Theorem 13.2.1 ensures that, within a given inner iteration, this matrix cannot become arbitrarily large, and therefore the induced norms still satisfy AN.1. We verify this property by examining the more general case where the trust-region shape is defined in the norm

$$\|\cdot\|_{M_{k,j}} = \sqrt{\langle \cdot, M_{k,j} \cdot \rangle}, \quad (13.7.2)$$

where

$$M_{k,j} \stackrel{\text{def}}{=} H_{k,j} + X_{k,j}^{-1} Z_{k,j} \quad (13.7.3)$$

and where we suppose that

$$\lambda_{\min}[M_{k,j}] \geq \epsilon \text{ and } \|H_{k,j}\| \leq \kappa_H \quad (13.7.4)$$

for some  $\epsilon \in (0, 1)$  and  $\kappa_H > 0$  independent of  $j$ .

**Theorem 13.7.1** Suppose that AC.1c, AC4.c, and AC.8 hold and that  $\{x_j\}$  is a sequence of iterates generated by Algorithm 13.3.1, where  $m_j^f$  and  $m_j^b$  satisfy AM.1–AM.3, AM.4h, and AM.4i. Suppose furthermore that  $M_{k,j}$  is defined by (13.7.3) and (13.7.4) and that AA.7 holds. Then the norms (13.7.2) satisfy AN.1 for  $k$  fixed.

**Proof.** First notice that both inequalities of AN.1 obviously hold for zero vectors. We therefore restrict our attention to vectors  $y \neq 0$ . Suppose first that

$$\langle y, H_{k,j} y \rangle \leq \langle y, X_{k,j}^{-1} Z_{k,j} y \rangle. \quad (13.7.5)$$

Then

$$\|y\|_{M_{k,j}}^2 = \langle y, [H_{k,j} + X_{k,j}^{-1} Z_{k,j}] y \rangle \leq 2\langle y, X_{k,j}^{-1} Z_{k,j} y \rangle = 2\frac{\kappa_{\text{uzi}}}{\kappa_{\text{mdb}}^2} \|y\|^2, \quad (13.7.6)$$

where we used AA.7 and Theorem 13.2.1. If, on the other hand, (13.7.5) does not hold, then

$$\|y\|_{M_{k,j}}^2 = \langle y, [H_{k,j} + X_{k,j}^{-1} Z_{k,j}] y \rangle \leq 2\langle y, H_{k,j} y \rangle \leq 2\kappa_H \|y\|^2, \quad (13.7.7)$$

because of the second part of (13.7.4). Combining (13.7.6) and (13.7.7), we obtain that

$$\min \left[ \frac{1}{2\kappa_{\text{uzi}}}, \frac{1}{2\kappa_H} \right] \kappa_{\text{mdb}}^2 \|y\|_{M_{k,j}}^2 \leq \|y\|^2. \quad (13.7.8)$$

Turning to the other requirement in AN.1, the first part of (13.7.4) implies that, for all  $y \neq 0$ ,

$$\frac{\|y\|^2}{\|y\|_{M_{k,j}}^2} = \frac{\langle M_{k,j}^{-\frac{1}{2}}(M_{k,j}^{\frac{1}{2}}y), M_{k,j}^{-\frac{1}{2}}(M_{k,j}^{\frac{1}{2}}y) \rangle}{\langle M_{k,j}^{\frac{1}{2}}y, M_{k,j}^{\frac{1}{2}}y \rangle} \leq \|M_{k,j}^{-1}\| \leq \frac{1}{\epsilon}.$$

This inequality and (13.7.8) together imply that AN.1 holds with

$$\kappa_{\text{une}} \stackrel{\text{def}}{=} \sqrt{\max \left[ \frac{1}{\epsilon}, \frac{2\kappa_{\text{uzi}}}{\kappa_{\text{mdb}}^2}, \frac{2\kappa_{\text{H}}}{\kappa_{\text{mdb}}^2} \right]}.$$

□

A result similar to this theorem holds under AM.4i for the primal method, where

$$M_{k,j} \stackrel{\text{def}}{=} H_{k,j} + \mu_k X_{k,j}^{-2}.$$

As a consequence, we see that the results of Theorems 13.6.2 and 13.6.4 are not affected by the use of the norm (13.7.2), (13.7.3).

The effect of using this norm is to “flatten” the “spherical” trust region of Figure 13.3.1 along the feasible region’s boundaries, as shown in Figure 13.7.1. This is desirable because one may wish to allow larger steps in directions corresponding to the null-space of the constraints. In particular, this has the benefit of allowing larger steps in directions of negative curvature in the null-space of the (nearly) active constraints. In practice, we may possibly have to modify  $\nabla_{xx} m_{k,j}(x_{k,j})$  (see Section 4.3.6) to define  $M_{k,j}$  so that the first part of (13.7.4) holds, while the second part is generally ensured by AM.4i.

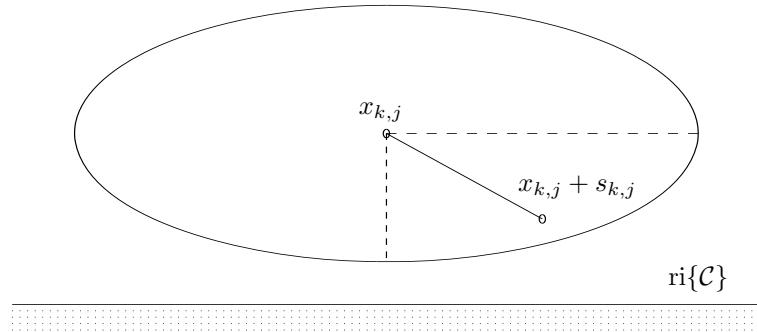


Figure 13.7.1: The shape of an anisotropic trust region in the neighbourhood of an active constraint.

### 13.7.3 Scaling of the Outer Iterations

We have just seen that scaling within inner iterations of the primal and primal-dual algorithms does not cause any particular difficulty. But scaling is also relevant to the outer iteration, because convergence will have been assessed *in the scaled space* when passing back from the inner to the outer iteration. This means that the norm used in the stopping tests (13.4.3)/(13.6.5) should be replaced by

$$\|\cdot\|_{k,j} \stackrel{\text{def}}{=} \|\cdot\|_{k+1} = \|\cdot\|_{M_{k+1}},$$

where

$$M_{k+1} \stackrel{\text{def}}{=} H_{k+1} + \mu_k X_{k+1}^{-2}$$

for the primal method, or

$$M_{k+1} \stackrel{\text{def}}{=} H_{k+1} + X_{k+1}^{-1} Z_{k+1}$$

for the primal-dual method, where  $H_{k+1} = H_{k,j}$  is, as above, a general bounded symmetric matrix and where  $M_{k+1}$  is uniformly positive definite. More precisely, the scaled version of the test (13.4.3) in the primal method should now be of the form

$$\|\nabla_x f(x_{k+1}) - \mu_k X_{k+1}^{-1} e\|_{[k+1]} \leq \epsilon^D(\mu_k), \quad (13.7.9)$$

and the test (13.6.5) for the primal-dual method should be replaced by

$$\|\nabla_x f(x_{k+1}) - z_{k+1}\|_{[k+1]} \leq \epsilon^D(\mu_k), \quad (13.7.10)$$

where  $\|\cdot\|_{[k+1]}$  is the dual norm of  $\|\cdot\|_{k+1}$ . The use of the dual norm is justified because (13.7.9) and (13.7.10) measure convergence of the gradient.<sup>218</sup> Using Lemma 8.1.6 (p. 260), we deduce that this dual norm is defined by

$$\|\cdot\|_{[k+1]} = \|\cdot\|_{M_{k+1}^{-1}}. \quad (13.7.11)$$

Similarly, the test (13.4.4)/(13.6.7) must now be performed in the scaled space. This means that, for the pure primal method, we must now ensure that

$$\lambda_{\min}[M_{k+1}^{-\frac{1}{2}}(\nabla_{xx} f(x_{k+1}) + \mu_k X_{k+1}^{-2})M_{k+1}^{-\frac{1}{2}}] \geq -\epsilon^E(\mu_k) \quad (13.7.12)$$

instead of (13.4.4), while the condition (13.6.7) should be replaced by

$$\lambda_{\min}[M_{k+1}^{-\frac{1}{2}}(\nabla_{xx} f(x_{k+1}) + X_{k+1}^{-1} Z_{k+1})M_{k+1}^{-\frac{1}{2}}] \geq -\epsilon^E(\mu_k) \quad (13.7.13)$$

for the primal-dual method.

At first sight, we may question whether the global convergence properties of the unscaled versions of the algorithms continue to hold if we use the scaled norms. The question arises because the matrices  $M_k$  blow up when, as is highly likely, the iterates approach the boundary of the feasible region. Thus the norms used in the outer iteration are no longer uniformly equivalent; that is, AN.1 is violated.

It is fortunate that global convergence to critical points may still be proved if the scaled formulations (13.7.9)/(13.7.10) and (13.7.12)/(13.7.13) are used. We have to reprove the conclusions of Theorem 13.6.3.

---

<sup>218</sup>See Section 8.1.4. For instance, if we use the preconditioned conjugate gradient method of Section 5.1.6 in the inner iteration, the trust region is then defined in the  $M_{k,j}$  norm and the residual is then measured in the  $M_{k,j}^{-1}$  norm; see Sections 7.4 and 7.5.

**Theorem 13.7.2** Let the assumptions of Theorem 13.6.3 hold, except that (13.6.5) is replaced by (13.7.10) and (13.6.7) by (13.7.13) in the definition of Algorithm 13.6.2. Suppose further that, for all  $k$ ,

$$\|\cdot\|_k = \|\cdot\|_{M_k},$$

where

$$M_k \stackrel{\text{def}}{=} H_k + X_k^{-1} Z_k,$$

and that

$$\lambda_{\min}[M_k] \geq \epsilon \text{ and } \|H_k\| \leq \kappa_H \quad (13.7.14)$$

for some  $\epsilon \in (0, 1)$  and  $\kappa_H > 0$  independent of  $k$ . Suppose finally that

$$\lim_{k \rightarrow \infty} \frac{\epsilon^C(\mu_k)}{\mu_k} \leq \kappa_\mu \quad (13.7.15)$$

for some  $\kappa_\mu > 0$  and that

$$\lim_{k \rightarrow \infty} \frac{\epsilon^D(\mu_k)\sqrt{\mu_k}}{\min_i[x_{k+1}]_i} = 0. \quad (13.7.16)$$

Then the conclusions of Theorem 13.6.3 remain valid.

**Proof.** We first introduce a notion that is akin to that of a consistently active subsequence in that we choose a subsequence indexed by  $\mathcal{K}$  such that

$$\lim_{k \rightarrow \infty} \frac{[z_k]_i}{[x_k]_i} = +\infty \quad (i \in \mathcal{E}) \quad \text{and} \quad \limsup_{k \rightarrow \infty} \frac{[z_k]_i}{[x_k]_i} < \infty \quad (i \in \mathcal{R}) \quad (13.7.17)$$

for some subsets  $\mathcal{E}$  and  $\mathcal{R}$  of  $\{1, \dots, n\}$ . The variables whose index is in  $\mathcal{E}$  converge quickly to their bound (they are “eager”), while those whose index is in  $\mathcal{R}$  are “reluctant” to do so, if they converge to their bound at all. As was the case for consistently active subsequences, note that the complete sequence of iterates may be partitioned into a finite set of subsequences satisfying (13.7.17) (for different sets  $\mathcal{E}$  and  $\mathcal{R}$ ). Let  $\kappa > 0$  be such that

$$\kappa \geq \max_{i \in \mathcal{R}} \limsup_{k \rightarrow \infty} \frac{[z_k]_i}{[x_k]_i}.$$

Writing  $r_k = \nabla_x f(x_k) - z_k$  and using (13.7.11), condition (13.7.10) then becomes

$$\|M_k^{-\frac{1}{2}} r_k\| \leq \epsilon^D(\mu_{k-1})$$

for all  $k \in \mathcal{K}$ . But, since  $M_k$  is positive definite,

$$\|M_k^{-\frac{1}{2}} r_k\|^2 \geq \frac{\|r_k\|^2}{\lambda_{\max}[M_k]} = \frac{\|r_k\|^2}{\|M_k\|}.$$

Now we have, using (13.7.14), that, for  $k \in \mathcal{K}$  sufficiently large,

$$\|M_k\| \leq \|H_k\| + \|X_k^{-1}Z_k\| \leq \kappa_H + \max_i \frac{[z_k]_i}{[x_k]_i}.$$

Assume first that  $\mathcal{E} = \emptyset$ . Then  $\|M_k\| \leq \kappa_H + 2\kappa$  for  $k \in \mathcal{K}$  sufficiently large, and therefore, using (13.7.10),

$$\epsilon^D(\mu_{k-1}) \geq \|M_k^{-\frac{1}{2}}r_k\| \geq \frac{\|r_k\|}{\sqrt{\kappa_H + 2\kappa}}$$

for such  $k$ . This implies that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \|r_k\| = \lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{K}}} \|\nabla_x f(x_k) - z_k\| = 0. \quad (13.7.18)$$

On the other hand, if  $\mathcal{E} \neq \emptyset$ ,

$$\|M_k\| \leq 2 \max_i \frac{[z_k]_i}{[x_k]_i} \leq 2 \max_i \left[ \frac{\mu_{k-1}}{[x_k]_i^2} + \frac{[X_k z_k - \mu_{k-1} e]_i}{[x_k]_i^2} \right] \leq 2(1 + \kappa_\mu) \frac{\mu_{k-1}}{\min_i [x_k]_i^2}$$

for  $k \in \mathcal{K}$  sufficiently large, where we have used (13.6.6) and (13.7.16) to deduce the third inequality. In this case,

$$\|M_k^{-\frac{1}{2}}r_k\| \geq \|r_k\| \frac{\min_i [x_k]_i}{\sqrt{2(1 + \kappa_\mu)\mu_{k-1}}},$$

and hence

$$\|r_k\| \leq \sqrt{2(1 + \kappa_\mu)} \frac{\epsilon^D(\mu_{k-1})\sqrt{\mu_{k-1}}}{\min_i [x_k]_i},$$

which, together with (13.7.16), again yields (13.7.18). Thus, because  $z_k \geq 0$  for all  $k$  and because  $\mathcal{K}$  is arbitrary, we obtain that (13.4.9) holds in all cases.

The final step of our proof is to show that (13.4.11) continues to hold, that is, that the Hessian of the objective function is asymptotically positive semidefinite along a consistently active subsequence. We first notice that (13.7.13), the forcing nature of  $\epsilon^E(\mu)$ , and the convergence of  $\mu_k$  to zero imply that

$$\liminf_{\ell \rightarrow \infty} \inf_{v \neq 0} \frac{\langle v, M_{k_\ell}^{-\frac{1}{2}} [\nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}] M_{k_\ell}^{-\frac{1}{2}} v \rangle}{\|v\|^2} \geq 0.$$

But, if we define  $w = M_{k_\ell}^{-\frac{1}{2}}v$ , we have that

$$\frac{\langle v, M_{k_\ell}^{-\frac{1}{2}} [\nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}] M_{k_\ell}^{-\frac{1}{2}} v \rangle}{\|v\|^2} = \frac{\langle w, [\nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}] w \rangle}{\|w\|_{k_\ell}^2},$$

where we have used the identity  $\|w\|_{k_\ell} = \|M_{k_\ell}^{\frac{1}{2}}w\|$  to derive the denominator of the right-hand side. Since  $M_{k_\ell}^{\frac{1}{2}}$  has full rank by assumption, we obtain that

$$\liminf_{\ell \rightarrow \infty} \tau_{k_\ell} = \liminf_{\ell \rightarrow \infty} \inf_{w \neq 0} \frac{\langle w, [\nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}] w \rangle}{\|w\|_{k_\ell}^2} \geq 0,$$

where  $\tau_k$  is defined as in AA.2b. We may then apply Lemma 8.1.1<sup>219</sup> (p. 256), where we have from the definition of the set  $\mathcal{K}_\epsilon$  in (8.1.6) (p. 253) and the first part of our proof that, for all  $\epsilon > 0$ ,  $\mathcal{K}_\epsilon = \text{span} \{e_i \mid i \in \mathcal{R}\} \stackrel{\text{def}}{=} \mathcal{N}$ , which is a subspace. This yields that

$$\liminf_{\ell \rightarrow \infty} \lambda_{\min} \left[ N^T [\nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}] N \right] \geq 0,$$

where  $N$  is a matrix whose columns form an orthonormal basis of  $\mathcal{N}$ . Observe now that the convergence of the inner iteration to a first-order critical point and AA.8 imply that

$$\lim_{\ell \rightarrow \infty} \|z_{k_\ell} - \mu_{k_\ell-1} X_{k_\ell}^{-1} e\| = 0,$$

and thus that

$$\liminf_{\ell \rightarrow \infty} \lambda_{\min} \left[ N^T [\nabla_{xx} f(x_{k_\ell}) + \mu_{k_\ell-1} X_{k_\ell}^{-2}] N \right] \geq 0. \quad (13.7.19)$$

As in the proof of Theorem 13.6.3, we now assume that (13.6.13) does not hold, which means that we can pick a unit vector  $u$  such that

$$[u]_i = 0 \text{ for } i \in \mathcal{A} \text{ and } \liminf_{\ell \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_\ell}) u \rangle = -\epsilon \quad (13.7.20)$$

for some  $\epsilon > 0$ . As for the unscaled case, the convergence of the barrier parameters to zero implies that

$$\lim_{\ell \rightarrow \infty} \mu_{k_\ell-1} X_{k_\ell}^{-2} e_i = 0$$

for  $i \notin \mathcal{A}$ , and the first part of (13.7.20) then gives that

$$\lim_{\ell \rightarrow \infty} \mu_{k_\ell-1} \langle u, X_{k_\ell}^{-2} u \rangle = 0.$$

Hence we deduce from this limit that

$$\liminf_{\ell \rightarrow \infty} \langle u, [\nabla_{xx} f(x_{k_\ell}) + \mu_{k_\ell-1} X_{k_\ell}^{-2}] u \rangle = \liminf_{\ell \rightarrow \infty} \langle u, \nabla_{xx} f(x_{k_\ell}) u \rangle = -\epsilon.$$

However, the first part of (13.7.20) and the fact that  $i \in \mathcal{R}$  if  $i \notin \mathcal{A}$  imply that  $u \in \mathcal{N}$ . Hence, since the columns of  $N$  are orthogonal, we have that  $u = Nh$  for some unit vector  $h$ . This then yields that

$$\liminf_{\ell \rightarrow \infty} \langle h, N^T [\nabla_{xx} f(x_{k_\ell}) + \mu_{k_\ell-1} X_{k_\ell}^{-2}] Nh \rangle = -\epsilon.$$

But this contradicts (13.7.19). Hence no vector satisfying (13.7.20) can exist, (13.4.11) holds, and the proof of the theorem is complete.  $\square$

A similar result can obviously be proved in the case of the primal algorithm. The scaling matrix  $M_k$  is then defined by

$$M_k = H_k + \mu_{k-1} X_k^{-2}. \quad (13.7.21)$$

---

<sup>219</sup>With  $\nabla_{xx} m_{k_\ell}(x_{k_\ell}) = \nabla_{xx} f(x_{k_\ell}) + X_{k_\ell}^{-1} Z_{k_\ell}$ .

Since  $z_k = \mu_{k-1} X_k^{-1} e$  is an admissible choice for the dual variables, the above proof applies to this case with the additional simplification that (13.7.16) is no longer needed because of the definition of  $z_k$ . For reference, we formally state the corresponding result.

**Theorem 13.7.3** Let the assumptions of Theorem 13.4.3 hold, except that (13.4.3) is replaced by (13.7.9) and (13.4.4) by (13.7.12) in the definition of Algorithm 13.4.1. Suppose also that

$$\|\cdot\|_k = \|\cdot\|_{M_k},$$

where  $M_k$  is now given by (13.7.21), and that (13.7.14) holds for some  $\epsilon \in (0, 1)$  and  $\kappa_H > 0$  independent of  $k$ . Suppose further that (13.7.15) also holds. Then the conclusions of Theorem 13.4.3 remain valid.

## Notes and References for Section 13.7

The analysis of the eigenstructure of Hessians arising in barrier methods is not a new subject. Interesting early contributions are those of Murray (1971b) and Lootsma (1969), who analyse the case where all dual variables associated with active constraints are assumed to be nonzero.

## 13.8 Upper and Lower Bounds on the Variables

The form of the problem considered so far in this chapter is of course somewhat restrictive, since we have only considered nonnegativity constraints on the variables. We are thus interested in trying to apply the same ideas to more general cases and shall discuss these extensions in this section and its successors. We start by extending our results on barrier methods to the case where, instead of nonnegativity constraints, we have general lower and upper bounds of the form

$$x_\ell \leq x \leq x_u,$$

where the inequalities are meant componentwise. In this case, the assumption AC.2b simply requires<sup>220</sup> that

$$x_\ell < x_u.$$

In fact, we simply have to rewrite the barrier terms corresponding to this new feasible region. This can easily be done: the new log-barrier term is then given by

$$b^{\log}(x, \mu) = -\mu \langle e, \log(x - x_\ell) + \log(x_u - x) \rangle, \quad (13.8.1)$$

<sup>220</sup>This is not a severe restriction, since it is only violated when one or more of the lower bounds matches its corresponding upper bound. In this case, the variables in question are fixed and can be removed from the problem formulation.

while the reciprocal barrier terms become

$$b^{R(\alpha)}(x, \mu) = \frac{\mu}{\alpha} \sum_{i=1}^n \left( \frac{1}{[x - x_\ell]_i^\alpha} + \frac{1}{[x_u - x]_i^\alpha} \right). \quad (13.8.2)$$

We permit components of  $x_\ell$  to be  $-\infty$  and components of  $x_u$  to be  $+\infty$ , which allows any combinations of bound constraints within our framework. If one of the bounds is infinite, then its corresponding term is deleted from the summations in (13.8.1) or (13.8.2).

In the preceding development, the fact that the constraints we were considering were simply nonnegativities was of little significance. As a consequence, all of the theoretical and algorithmic developments concerning primal and primal-dual log-barrier algorithms extend without modifications to the more general case involving simple bounds. However, we must remember that there are now dual variables for every finite bound, which means that we may have up to  $2n$  dual variables in the primal-dual algorithm. If all of the bounds are finite, the feasible region  $\mathcal{C}$  is bounded, and we no longer need to assume that the barrier functions are bounded below (in Theorems 13.4.2 and 13.6.2), because this is automatically guaranteed by the continuity of all functions involved. Furthermore, the sequences of iterates generated by our algorithms remain in a bounded domain, since we have insisted on keeping each iterate strictly feasible. Thus these sequences always have limit points, which simplifies Theorems 13.4.3 and 13.6.3. The same is true if we assume AI.1.

## 13.9 Barrier Methods for General Constraints

The situation is essentially the same if we now consider the general problem (13.1.3), where  $\mathcal{C}$  is a closed convex set with nonempty relative interior. There is still, however, a need for us to be able to express a barrier term for this set. This is relatively simple when AC.7 holds, that is, when

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid c_i(x) \geq 0 \text{ for } i = 1, \dots, m\}, \quad (13.9.1)$$

where the feasible region  $\mathcal{C}$  is assumed to satisfy AC.1 (the constraint functions are twice-continuously differentiable), AC.4 (they have bounded Hessians), and the following additional requirements.

**AC.2c** The feasible region has a nonempty *strict interior*  $\text{si}\{\mathcal{C}\}$ ; that is,

$$\text{si}\{\mathcal{C}\} \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid c_i(x) > 0 \text{ for all } i = 1, \dots, m\} \neq \emptyset.$$

**AC.5** There exists a constant  $\kappa_{\text{ubgc}} > 0$  such that

$$\|\nabla_x c_i(x)\| \leq \kappa_{\text{ubgc}}$$

for all  $x \in \mathcal{C}$  and every  $i \in \{1, \dots, m\}$ .

The latter assumption requires that the Jacobian  $A(x)$  of the constraints remains bounded in the feasible domain. Given such a  $\mathcal{C}$ , (13.8.1) and (13.8.2) become

$$b^{\log}(x, \mu) = -\mu \langle e, \log(c(x)) \rangle \text{ and } b^{R(\alpha)}(x, \mu) = \sum_{i=1}^m \frac{\mu}{\alpha[c_i(x)]^\alpha},$$

respectively. If  $C(x) = \text{diag}[c_1(x), \dots, c_m(x)]$ , their derivatives are given by

$$\nabla_x b^{R(\alpha)}(x, \mu) = -\mu A^T(x) C^{-(\alpha+1)}(x) e$$

and

$$\nabla_{xx} b^{R(\alpha)}(x, \mu) = \mu \left[ (\alpha + 1) A(x)^T C^{-(\alpha+2)}(x) A(x) - \sum_{i=1}^m \frac{1}{[c_i(x)]^{\alpha+1}} \nabla_{xx} c_i(x) \right],$$

where  $\alpha = 0$  corresponds to the log-barrier case and where  $A(x)$  is the Jacobian of the constraints. Appropriate models for the log-barrier and reciprocal barrier are then

$$b^{\log}(x, \mu) - \mu \langle C^{-1}(x)e, A(x)s \rangle + \frac{1}{2}\mu \langle A(x)s, C^{-2}(x)A(x)s \rangle - \frac{1}{2} \sum_{i=1}^m \frac{\mu}{c_i(x)} \langle s, \nabla_{xx} c_i(x)s \rangle$$

and

$$\begin{aligned} b^{R(\alpha)}(x, \mu) - \mu \langle C^{-(\alpha+1)}(x)e, A(x)s \rangle + \frac{1}{2}\mu(\alpha + 1) \langle A(x)s, C^{-(\alpha+2)}(x)A(x)s \rangle \\ - \frac{1}{2} \sum_{i=1}^m \frac{\mu}{[c_i(x)]^{\alpha+1}} \langle s, \nabla_{xx} c_i(x)s \rangle \end{aligned}$$

for  $\alpha \geq 1$ , while the direct extension of the perturbation argument of Section 13.6 shows that the primal-dual model is now given by

$$b^{\log}(x, \mu) - \mu \langle C^{-1}(x)e, A(x)s \rangle + \frac{1}{2} \langle A(x)s, [C^{-1}(x)Y]A(x)s \rangle - \frac{1}{2} \sum_{i=1}^m [y]_i \langle s, \nabla_{xx} c_i(x)s \rangle.$$

Note, as before, how the models for the log-barrier and primal-dual cases coincide if we set dual variables  $y = \mu C^{-1}(x)e$ . We now review how the primal-dual method<sup>221</sup> should be restated when the constraints have the form (13.9.1). We first consider the inner iterations, whose aim is now to (approximately) minimize

$$\phi^{\log}(x, \mu) = f(x) - \mu \langle e, \log(c(x)) \rangle \quad (13.9.2)$$

using the model of the barrier term given above. That is, at iteration  $(k, j)$ ,

$$m_{k,j}(x_{k,j} + s_{k,j}) = m_{k,j}^f(x_{k,j} + s_{k,j}) + m_{k,j}^b(x_{k,j} + s_{k,j}), \quad (13.9.3)$$

where

$$\begin{aligned} m_{k,j}^b(x_{k,j} + s_{k,j}) &= -\mu_k \langle e, \log(c(x_{k,j})) \rangle - \mu_k \langle C^{-1}(x_{k,j})e, A(x_{k,j})s_{k,j} \rangle \\ &\quad + \frac{1}{2} \langle A(x_{k,j})s_{k,j}, [C^{-1}(x_{k,j})Y_{k,j}]A(x_{k,j})s_{k,j} \rangle \\ &\quad - \frac{1}{2} \sum_{i=1}^m [y_{k,j}]_i \langle s_{k,j}, \nabla_{xx} c_i(x_{k,j})s_{k,j} \rangle. \end{aligned} \quad (13.9.4)$$

---

<sup>221</sup>Once again, the purely primal version is recovered by setting  $y = \mu C^{-1}(x)e$ .

We now state the corresponding primal-dual inner algorithm.<sup>222</sup>

**Algorithm 13.9.1: Inner primal-dual algorithm for general constraints**

**Step 0: Initialization.** An initial point  $x_{k,0} \in \text{si}\{\mathcal{C}\}$ , a vector  $y_{k,0} > 0$  of dual variables, and an initial trust-region radius  $\Delta_{k,0}$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Finally, the constant  $\varsigma_k \in (0, 1)$  is also given. Set  $j = 0$ .

**Step 1: Model definition.** Define a model  $m_{k,j}$  of (13.9.2) of the form (13.9.3), (13.9.4) in  $\mathcal{B}_{k,j}$ , where  $m_{k,j}^f$  is a model of  $f(x_{k,j} + s)$  satisfying AM.1–AM.3 and AM.4i.

**Step 2: Step calculation.** Compute a step  $s_{k,j}$  that sufficiently reduces the model  $m_{k,j}$  in the sense of AA.1/AA.2 and such that  $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$ .

**Step 3: Acceptance of the trial point.** If  $c_i(x_{k,j} + s_{k,j}) \geq \varsigma_k c_i(x_{k,j})$  for  $i = 1, \dots, m$ , compute  $\phi(x_{k,j} + s_{k,j}, \mu_k)$  and define the ratio

$$\rho_{k,j} = \frac{\phi(x_{k,j}, \mu_k) - \phi(x_{k,j} + s_{k,j}, \mu_k)}{m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j})};$$

else set  $\rho_{k,j} = -\infty$ . Then if  $\rho_{k,j} \geq \eta_1$ , define  $x_{k,j+1} = x_{k,j} + s_{k,j}$ ; otherwise define  $x_{k,j+1} = x_{k,j}$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k,j+1} \in \begin{cases} [\Delta_{k,j}, \infty) & \text{if } \rho_{k,j} \geq \eta_2, \\ [\gamma_2 \Delta_{k,j}, \Delta_{k,j}] & \text{if } \rho_{k,j} \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_{k,j}, \gamma_2 \Delta_{k,j}] & \text{if } \rho_{k,j} < \eta_1. \end{cases}$$

Increment  $j$  by 1 and go to Step 1.

Note that this is a primal-dual version of Algorithm 13.2.1, in that the requirement that the step remain feasible is imposed *after* the trial step is computed. The convergence properties of the primal-dual inner algorithm naturally extend to this new algorithm. All concepts defined for bounds, such as that of consistently active subsequences, extend in the obvious way. Note that AA.7 must be replaced by the condition that

$$[y_{k,j}]_i \leq \kappa_{\text{uby}} \max \left[ \frac{1}{c_i(x_{k,j})}, 1 \right] \quad (13.9.5)$$

for all  $j \geq 0$  and for some  $\kappa_{\text{uby}} > 0$ , which, together with AM.4h, AC.1, and AC.4, ensure the boundedness of the Hessian of the objective function (13.9.2) in  $\text{si}\{\mathcal{C}\}$ , as required by AM.4i.

---

<sup>222</sup>For easier comparison, we have again assumed that  $\|\cdot\|_{k,j} = \|\cdot\|$  for all  $k$  and  $j$ .

Using this inner algorithm, we then expect to reach an approximate second-order critical point of the problem of minimizing  $\phi^{\log}(x, \mu_k)$  subject to  $c(x) \geq 0$ , because the convergence of the algorithm ensures that

$$\lim_{j \rightarrow \infty} \|\nabla_x f(x_{k,j}) - A^T(x_{k,j})y_{k,j}\| = \lim_{j \rightarrow \infty} \|C(x_{k,j})Y_{k,j}e - \mu_k I\| = 0.$$

Furthermore

$$\limsup_{j \rightarrow \infty} \lambda_{\min}[\nabla_{xx}\ell(x_{k,j}, y_{k,j}) + G_{k,j}] \geq 0,$$

where

$$\nabla_{xx}\ell(x_{k,j}, y_{k,j}) = \nabla_{xx}f(x_{k,j}) - \sum_{i=1}^m [y_{k,j}]_i \nabla_{xx}c_i(x_{k,j})$$

is the Hessian of the Lagrangian function associated with the problem and where

$$G_{k,j} \stackrel{\text{def}}{=} A^T(x_{k,j})C^{-1}(x_{k,j})Y_{k,j}A(x_{k,j}),$$

while, for all  $j$ ,  $(c(x_{k,j}), y_{k,j}) > 0$ . Given that these properties hold at limit points of the inner algorithm, we may then use them in a relaxed form to define suitable stopping criteria for the inner iterations, which then implies that the outer algorithm takes the following form.

**Algorithm 13.9.2: Primal-dual barrier algorithm for general constraints**

**Step 0: Initialization.** An initial point  $x_0$  such that  $c(x_0) > 0$ , a vector of dual variables  $y_0 > 0$ , and an initial barrier parameter  $\mu_0 > 0$  are given. The forcing functions  $\epsilon^C(\mu)$ ,  $\epsilon^D(\mu)$ , and  $\epsilon^E(\mu)$  are also given. Set  $k = 0$ .

**Step 1: Inner minimization.** Choose a value  $\varsigma_k \in (0, 1)$ . Approximately minimize the log-barrier function  $\phi^{\log}(x, \mu_k) = f(x) - \mu_k \langle e, \log(c(x)) \rangle$  starting from  $x_k$  and using Algorithm 13.9.1. Stop the algorithm as soon as an iterate  $(x_{k,j}, y_{k,j}) = (x_{k+1}, y_{k+1})$  is found such that

$$\|\nabla_x f(x_{k+1}) - A^T(x_{k+1})y_{k+1}\| \leq \epsilon^D(\mu_k), \quad (13.9.6)$$

$$\|C(x_{k+1})Y_{k+1}e - \mu_k I\| \leq \epsilon^C(\mu_k), \quad (13.9.7)$$

$$\lambda_{\min}[\nabla_{xx}\ell(x_{k+1}, y_{k+1}) + G_{k+1}] \geq -\epsilon^E(\mu_k), \quad (13.9.8)$$

and

$$(c(x_{k+1}), y_{k+1}) \geq 0.$$

**Step 2: Update the barrier parameter.** Choose  $\mu_{k+1} > 0$  in such a way as to ensure that  $\lim_{k \rightarrow \infty} \mu_k = 0$ . Increment  $k$  by 1 and return to Step 1.

It remains to verify that this latter algorithm produces a sequence of points at which weak second-order criticality conditions are asymptotically satisfied. Although the proof of this result is entirely similar in spirit to that of Theorem 13.6.3, we give it in detail for the sake of clarity.

**Theorem 13.9.1** Suppose that AF.1–AF.3 hold for the objective function  $f(x)$  and that the model  $m_{k,j}^f$  of the objective function satisfies AM.1–AM.3, AM.4i, AM.5c, and AM.6. Suppose also that AW.2 holds for  $\phi^{\log}(x, \mu)$  and for all  $\mu > 0$ , and that (13.9.5) holds. If  $\{x_k\}$  is a sequence of iterates generated by Algorithm 13.9.2, then we have that

$$\lim_{k \rightarrow \infty} \|\nabla_x f(x_k) - A^T(x_k)y_k\| = 0 \text{ and } \lim_{k \rightarrow \infty} C(x_k)Y_k e = 0. \quad (13.9.9)$$

If we suppose further that  $\{x_{k_p}\}$  is a consistently active subsequence of iterates and that

$$\lim_{j \rightarrow \infty} \|y_{k_p} - \mu_{k_p-1}C^{-1}(x_{k_p})e\| = 0 \quad (13.9.10)$$

if (13.9.9) holds, then we also have that

$$\liminf_{p \rightarrow \infty} \inf_{u \in \mathcal{N}_{\mathcal{A}}(x_{k_p})} \langle u, \nabla_{xx} \ell(x_{k_p}, y_{k_p})u \rangle \geq 0, \quad (13.9.11)$$

where  $\mathcal{N}_{\mathcal{A}}(x_{k_p})$  is the null-space of the submatrix of the Jacobian  $A(x_{k_p})$  consisting of the rows of this matrix whose index is in  $\mathcal{A}(x_{k_p})$ .

**Proof.** First observe that the convergence of  $\mu_k$  to zero, the forcing nature of the functions  $\epsilon^D(\mu)$  and  $\epsilon^C(\mu)$ , and the conditions (13.9.6), (13.9.7) ensure that (13.9.9) holds.

Let us now consider  $\{x_{k_p}\}$ , a consistently active subsequence of iterates (with respect to the constraints (13.9.1)) with an associated active set  $\mathcal{A} = \mathcal{A}\{x_{k_p}\}$ , and suppose that (13.9.11) does not hold. Then there must exist a sequence  $\{u_{k_p}\}$  of unit vectors such that

$$u_{k_p} \in \mathcal{N}_{\mathcal{A}}(x_{k_p}) \text{ and } \lim_{p \rightarrow \infty} \langle u_{k_p}, \nabla_{xx} \ell(x_{k_p}, y_{k_p})u_{k_p} \rangle = -\epsilon \quad (13.9.12)$$

for some  $\epsilon > 0$ . We obtain, from (13.9.10), that

$$\lim_{p \rightarrow \infty} \langle u_{k_p}, G_{k_p}u_{k_p} \rangle = \lim_{p \rightarrow \infty} \mu_{k_p-1} \|C^{-1}(x_{k_p})A(x_{k_p})u_{k_p}\|^2. \quad (13.9.13)$$

If we now define  $A_{k_p}^{\mathcal{A}}$  to be the  $m \times n$  matrix constructed from  $A(x_{k_p})$  by setting its rows whose index in not in  $\mathcal{A}$  to zero, and define  $A_{k_p}^{\mathcal{F}}$  to be its complement, that is,  $A(x_{k_p}) = A_{k_p}^{\mathcal{A}} + A_{k_p}^{\mathcal{F}}$ , we have that, by construction and the first part of (13.9.12),

$$A_{k_p}^{\mathcal{A}}u_{k_p} = 0 \quad (13.9.14)$$

for every  $p$ . Furthermore, we also obtain that

$$\|A_{k_p}^{\mathcal{F}}\| \leq \|A(x_{k_p})\| \leq \kappa_{\text{ubgc}},$$

because of the definition of  $A_{k_p}^{\mathcal{F}}$  and AC.5. If we now use this latter bound, the identity  $\|u_{k_p}\| = 1$ , the fact that  $c_i(x_{k_p})$  is bounded away from zero for  $i \notin \mathcal{A}$ , and the convergence of  $\mu_{k_p-1}$  to zero, we derive that

$$\lim_{p \rightarrow \infty} \mu_{k_p-1} C^{-1}(x_{k_p}) A_{k_p}^{\mathcal{F}} u_{k_p} = 0.$$

Combining this limit with (13.9.14) and taking into account the orthogonality of the vectors  $C^{-1}(x_{k_p}) A_{k_p}^{\mathcal{F}} u_{k_p}$  and  $C^{-1}(x_{k_p}) A_{k_p}^{\mathcal{A}} u_{k_p}$ , we deduce that

$$\begin{aligned} \lim_{p \rightarrow \infty} \mu_{k_p-1} \|C^{-1}(x_{k_p}) A(x_{k_p}) u_{k_p}\|^2 &= \lim_{p \rightarrow \infty} \mu_{k_p-1} \|C^{-1}(x_{k_p}) A_{k_p}^{\mathcal{A}} u_{k_p}\|^2 \\ &\quad + \lim_{p \rightarrow \infty} \mu_{k_p-1} \|C^{-1}(x_{k_p})_{k_p}^{\mathcal{F}} u_{k_p}\|^2 \\ &= 0. \end{aligned}$$

Consequently, in view of (13.9.13), it follows that

$$\lim_{p \rightarrow \infty} \langle u_{k_p}, G_{k_p} u_{k_p} \rangle = 0.$$

Hence we deduce from this last limit and the second part of (13.9.12) that

$$\lim_{p \rightarrow \infty} \langle u_{k_p}, [\nabla_{xx} \ell(x_{k_p}, y_{k_p}) + G_{k_p}] u_{k_p} \rangle = \lim_{p \rightarrow \infty} \langle u_{k_p}, \nabla_{xx} \ell(x_{k_p}, y_{k_p}) u_{k_p} \rangle = -\epsilon,$$

which, given the forcing nature of  $\epsilon^E(\mu)$  and the convergence of  $\mu_k$  to zero, contradicts (13.9.8) for  $\ell$  sufficiently large. Thus (13.9.11) holds.  $\square$

Remember that the algorithm itself guarantees that  $(c(x_k), y_k) > 0$  for all  $k$ . Note also that (13.9.10) is the natural extension of AA.8 for the case where general constraints are present.

An easy corollary of this theorem is that, if the sequence  $(x_k, y_k)$  has a limit point  $(x_*, y_*)$ , then this limit point satisfies the first-order and weak second-order necessary conditions, as was already the case for the primal-dual barrier algorithm.

We conclude the section by an important observation. Note that convexity of the feasible domain has played very little role in the design and theory of the algorithms presented. In fact, its only use is to guarantee that the segment  $[x_{k,j}, x_{k,j} + s_{k,j}]$  lies entirely within  $\mathcal{C}$  (in the proof of Theorem 13.2.2). Since every iterate belongs to the strict interior of the feasible domain, the same guarantee can clearly be obtained for problems with nonconvex inequality constraints by assuming that the trust-region radius is small enough. Moreover, as we have already remarked, the convergence theory developed above only uses values of the trust-region radius that are “sufficiently small”. This means that the *algorithms of this chapter* (primal, primal-dual barrier, and the

variants we shall shortly consider that handle linear equality constraints) *can immediately be extended to the case of general, possibly nonconvex, inequality constraints*, which is a remarkable result.<sup>223</sup>

## Notes and References for Section 13.9

Our adaptation of the primal-dual algorithm and associated convergence theory to handle general constraints is based on Conn, Gould, Orban, and Toint (2000). The version of the algorithm discussed in this paper handles linear equality constraints (as in the next section) in addition to general nonconvex constraints and uses the scaling technique of Section 13.7. It is shown in Gould, Orban, Sartenaer, and Toint (2000a) that the inner iteration asymptotically reduces to a single step from  $x - k$  in order to compute  $x_{k,0}$ , and that the rate of convergence is nearly Q-quadratic.

We conclude this note by reminding the reader that the SUMT (sequential unconstrained minimization technique) of Fiacco and McCormick (1968) is the oldest barrier method that is provably convergent for problems with nonconvex objectives and nonconvex constraints. The behaviour of this famous algorithm was recently revisited by Nash and Sofer (1998).

## 13.10 Adding Linear Equality Constraints

Having considered how to cope with general inequality constraints, we now investigate the situation where linear equality constraints are present alongside nonnegativity requirements on the variables. In other words, we now consider that the feasible domain is described by

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax = b \text{ and } x \geq 0\}, \quad (13.10.1)$$

where  $A$  is an  $m \times n$  matrix and  $b \in \mathbb{R}^m$ . The set  $\mathcal{C}$  is thus nothing but the intersection of the affine subspace (hyperplane) defined by  $Ax = b$  with the nonnegative orthant  $\mathcal{O}_+$ . We will denote

$$\mathcal{C}_{\mathcal{E}} = \{x \in \mathbb{R}^n \mid Ax = b\},$$

so that  $\mathcal{C} = \mathcal{C}_{\mathcal{E}} \cap \mathcal{O}_+$ . Of course, we have to assume that feasible points exist for the problem to be well defined; that is, AC.2b holds. This assumption makes it possible to think about interior-point methods, since  $\mathcal{C}$  is required to have a (relative) interior.

The main idea is to apply the methods that we have discussed above for the case where nonnegativity, bound, or (linear) convex inequality constraints are present *in the affine subspace*  $\mathcal{C}_{\mathcal{E}}$ . Thus, if we are given an  $x_0 \in \text{ri}\{\mathcal{C}\}$ , we define a new variable  $v \in \mathbb{R}^p$  ( $p \geq n - m$ ) by

$$x = x_0 + Nv \quad (13.10.2)$$

for all  $x \in \mathcal{C}_{\mathcal{E}}$ , where  $N$  is a matrix of dimension  $n \times p$  whose columns form an orthonormal basis for the null-space  $\mathcal{N}$  of  $A$  (see Section 4.4.2). We note that  $v$  is, for now, a purely theoretical construct, and we will see below that we need not compute

---

<sup>223</sup>Which suggests that, maybe, this chapter should not belong in this part of the book, but rather in Part IV.

it in practice, nor do we need to know  $N$ . However, we pursue our idea by defining, for  $x$  and  $v$  satisfying (13.10.2),  $f^{\mathcal{C}_\varepsilon}(v) \stackrel{\text{def}}{=} f(x_0 + Nv) = f(x)$ , which then yields that

$$\nabla_v f^{\mathcal{C}_\varepsilon}(v) = N^T \nabla_x f(x_0 + Nv) = N^T \nabla_x f(x) \quad (13.10.3)$$

and

$$\nabla_{vv} f^{\mathcal{C}_\varepsilon}(v) = N^T \nabla_{xx} f(x_0 + Nv) N = N^T \nabla_{xx} f(x) N. \quad (13.10.4)$$

Our problem of minimizing  $f$  on  $\mathcal{C}$  now reduces to minimizing  $f^{\mathcal{C}_\varepsilon}(v)$  subject to

$$x_0 + Nv \geq 0. \quad (13.10.5)$$

The discussion of Section 13.9 then shows that, for instance, a primal-dual algorithm may be applied to this transformed problem, with convergence results given by Theorem 13.9.1, where we identify the constraint  $c(v) \geq 0$  with (13.10.5). Note that these results do *not* depend on  $A$  having full rank, but rather on our ability to define  $N$ . Nor is there any assumption on the values of the Lagrange multipliers associated with the constraint  $Ax = b$  in the original problem. Furthermore, Section 13.7 indicates that iteration-dependent scaling may be applied provided the norms used during the inner iterations satisfy AN.1. It is important to observe that these norms need only be defined in  $\mathcal{C}_\varepsilon$ , which is why we denote them by the symbol  $\|\cdot\|_{\mathcal{C}_\varepsilon,k}$ .

However, this approach may be inconvenient for two reasons. The first is that it requires the availability of  $N$ , which could be awkward when the dimension of the problem is large (see Section 4.4.2). Even if one is content to settle for the easier choice of a nonorthonormal basis of  $\mathcal{N}$ , the computational expense may still be substantial. Moreover this has the drawback of introducing the issue of the conditioning of this basis, which is the main reason why we shall not pursue this possibility any further. The second reason is the exchange of highly structured nonnegativity constraints for the less structured linear inequalities (13.10.5).

Thus there is some justification for trying to consider the problem in its original form within  $\mathbb{R}^n$  and to define an algorithm expressed in this space. The central feature of the approach is retained, in that we shall still insist on (strict) feasibility of all iterates with respect to the inequalities. The algorithm that we consider here is very similar to the primal-dual barrier algorithm (Algorithm 13.6.2). The inner iteration is carried out by a variant of the primal-dual inner algorithm (Algorithm 13.6.1), where Step 2 is replaced by the following.

**Algorithm 13.10.1: Step 2 of the constrained primal-dual inner algorithm**

**Step 2: Step calculation.** Define  $d_{k,j} = \text{dist}(x_{k,j}, \partial\mathcal{O}_+)$ . Compute a step  $s_{k,j}$  that “sufficiently reduces” the model  $m_{k,j}^{\text{PD}}$ , such that  $x_{k,j} + s_{k,j} \in \mathcal{B}_{k,j}$  and

$$As_{k,j} = 0, \quad \text{and } \text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{O}_+) \geq \varsigma_k d_{k,j}. \quad (13.10.6)$$

(We could equally consider a variant of the first primal-dual inner algorithm for the inner iteration, or of the primal barrier algorithm for the outer one.) The outer iteration is then given by a variant of the primal-dual barrier algorithm, where the constraint

$$Ax_k = b \quad (13.10.7)$$

is added to the conditions (13.6.5)–(13.6.8). We now verify that this procedure satisfies the conditions required for the primal-dual barrier method in  $\mathcal{C}_\varepsilon$  for the convergence results of Theorems 13.4.3 and 13.6.3. We first note that (13.10.6) and AC.5 ensure that  $x_{k,j} \in \text{ri}\{\mathcal{C}\}$  for all  $k$  and  $j$ , which implies that (13.10.7) is in fact redundant. The second observation is that the third part of (13.10.6), expressed here in terms of distance in  $\mathbb{R}^n$ , also ensures an equivalent inequality for the distance expressed in  $\mathcal{C}_\varepsilon$  when the iterates approach the boundary of the feasible set, as we now show.

**Lemma 13.10.1** Suppose that  $\mathcal{C}$  is given by (13.10.1) and that the columns of  $N$  form an orthonormal basis for the null-space of  $A$ . Let  $x$  be any vector in the relative interior of  $\mathcal{C}$  and denote the distance from  $x$  to the closest bound constraint in  $\mathcal{C}_\varepsilon$  by  $\text{dist}^{\mathcal{C}_\varepsilon}(x, \partial\mathcal{C})$ . Then there exists a  $\varsigma^{\mathcal{C}_\varepsilon} \in (0, 1)$  such that

$$\text{dist}^{\mathcal{C}_\varepsilon}(x, \partial\mathcal{C}) \geq \text{dist}(x, \partial\mathcal{O}_+) \geq \varsigma^{\mathcal{C}_\varepsilon} \min[1, \text{dist}^{\mathcal{C}_\varepsilon}(x, \partial\mathcal{C})]. \quad (13.10.8)$$

**Proof.** If we define  $v \in \mathbb{R}^p$  by

$$x = x_0 + Nv, \quad (13.10.9)$$

we first note that

$$\text{dist}^{\mathcal{C}_\varepsilon}(x, \partial\mathcal{C}) = \min_{\substack{i=1,\dots,n \\ w|[x_0+Nw]_i=0}} \|w - v\|. \quad (13.10.10)$$

If we now fix  $i \in \{1, \dots, n\}$ , the Lagrangian of the resulting reduced problem is given by

$$\ell_i(w, y_i) = \frac{1}{2} \|w - v\|^2 - y_i \langle e_i, x_0 + Nw \rangle,$$

where  $y_i$  is the Lagrange multiplier associated with the problem's single constraint. Equating  $\nabla_w \ell_i(w, y_i)$  to zero, we obtain that  $w_i$ , the solution of (13.10.10) for fixed  $i$ , is given by

$$w_i = v + y_i N^T e_i. \quad (13.10.11)$$

Substituting this value in the constraint and using (13.10.9), we deduce that

$$y_i = -\frac{\langle e_i, x \rangle}{\|N^T e_i\|^2} \quad (13.10.12)$$

if  $N^T e_i \neq 0$ . On the other hand, if  $N^T e_i = 0$ , then there is no point of the form (13.10.11) that satisfies the constraint, and this  $i$  therefore never defines  $\text{dist}(x, \partial\mathcal{C})$  for any  $x \in \mathcal{C}_\varepsilon$ . Thus we may restrict our attention to the subset of indices  $i$  such

that  $N^T e_i \neq 0$ . Note that there must be at least one such  $i$ , otherwise  $N$  would be the zero matrix, which is impossible. For such an  $i$ , we thus obtain from (13.10.11) and (13.10.12) that

$$\|w_i - v\| = \frac{\langle e_i, x_{k,j} \rangle}{\|N^T e_i\|},$$

which then yields that

$$\text{dist}^{\mathcal{C}_E}(x, \partial\mathcal{C}) = \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} \frac{[x]_i}{\|N^T e_i\|}.$$

But for each  $i$  such that  $N^T e_i \neq 0$ , we have that  $\|N^T e_i\| \in (0, 1]$  because  $e_i$  and the columns of  $N$  are normalized. This immediately gives that

$$\text{dist}(x, \partial\mathcal{O}_+) = \min_{i=1,\dots,n} [x]_i \leq \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} [x]_i \leq \text{dist}^{\mathcal{C}_E}(x, \partial\mathcal{C}),$$

and the first inequality of (13.10.8) follows. Assume now that

$$\min_{i=1,\dots,n} [x]_i < \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} [x]_i, \quad (13.10.13)$$

which means that  $x$  is closest to the bound that is parallel to  $\mathcal{C}_E$ . Then we have that

$$\text{dist}^{\mathcal{C}_E}(x, \partial\mathcal{C}) > \text{dist}(x, \partial\mathcal{O}_+) = \min_{\substack{i=1,\dots,n \\ N^T e_i = 0}} [x]_i = \varsigma_0, \quad (13.10.14)$$

where  $\varsigma_0$  is independent of  $x$  because all points in  $\mathcal{C}_E$  are at the same distance ( $\varsigma_0$ ) from the bound which achieves the minimum. On the other hand, if (13.10.13) does not hold, then

$$\begin{aligned} \text{dist}(x, \partial\mathcal{O}_+) &= \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} [x_{k,j}]_i \\ &\geq \left[ \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} \frac{[x_{k,j}]_i}{\|N^T e_i\|} \right] \left[ \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} \|N^T e_i\| \right] \\ &\stackrel{\text{def}}{=} \varsigma_1 \text{dist}^{\mathcal{C}_E}(x, \partial\mathcal{C}), \end{aligned} \quad (13.10.15)$$

where

$$\varsigma_1 \stackrel{\text{def}}{=} \min_{\substack{i=1,\dots,n \\ N^T e_i \neq 0}} \|N^T e_i\| \in (0, 1]$$

is a constant depending only on the geometry of  $\mathcal{C}$ . The second part of (13.10.8) then follows from the middle part of (13.10.14) and (13.10.15) with  $\varsigma^{\mathcal{C}_E} \stackrel{\text{def}}{=} \min[\varsigma_0, \varsigma_1]$ .

□

**Theorem 13.10.2** Suppose that  $\mathcal{C}$  is given by (13.10.1), that the third part of (13.10.6) holds, and that the columns of  $N$  form an orthonormal basis for the null-space of  $A$ . Let  $d_{k,j}^{\mathcal{C}_\varepsilon}$  denote the distance from  $x_{k,j}$  to the boundary of the feasible domain. Then there exists  $\varsigma_k^{\mathcal{C}_\varepsilon} \in (0, 1)$  such that

$$\text{dist}^{\mathcal{C}_\varepsilon}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k^{\mathcal{C}_\varepsilon} d_{k,j}^{\mathcal{C}_\varepsilon}$$

for  $d_{k,j}^{\mathcal{C}_\varepsilon}$  sufficiently small.

**Proof.** Suppose that

$$d_{k,j}^{\mathcal{C}_\varepsilon} < \varsigma_k^{\mathcal{C}_\varepsilon},$$

where  $\varsigma^{\mathcal{C}_\varepsilon}$  is given by the preceding lemma. This implies that the minimum must be achieved by the second argument of the minimum in the last right-hand side of (13.10.8). (In this case,  $d_{k,j}$  is not determined by the distance to a bound that is parallel to  $\mathcal{C}_\varepsilon$ .) As a consequence, we have that

$$\text{dist}^{\mathcal{C}_\varepsilon}(x_{k,j} + s_{k,j}, \partial\mathcal{C}) \geq \varsigma_k \text{dist}(x_{k,j} + s_{k,j}, \partial\mathcal{O}_+) \geq \varsigma_k \varsigma^{\mathcal{C}_\varepsilon} d_{k,j}^{\mathcal{C}_\varepsilon},$$

and we obtain the desired conclusion with  $\varsigma_k^{\mathcal{C}_\varepsilon} = \varsigma_k \varsigma^{\mathcal{C}_\varepsilon} \in (0, 1)$ .  $\square$

Of course, the fact that the conclusion of the theorem is only valid for  $d_{k,j}^{\mathcal{C}_\varepsilon}$  sufficiently small has no impact on the convergence theory, as this bound merely ensures that the trial points  $x_{k,j} + s_{k,j}$  cannot become arbitrarily close to the boundary of the feasible region (see footnote 206 on p. 502).

Two issues remain to be clarified. The first is to translate the “sufficient model reduction” that we imposed in  $\mathcal{C}_\varepsilon$  as a requirement in  $\mathbb{R}^n$ . The second is to examine what assuming AN.1 in  $\mathcal{C}_\varepsilon$  means in  $\mathbb{R}^n$ . Regarding this last issue, the main observation is that we only need  $\|\cdot\|_k$  to define a norm on  $\mathcal{N}$  and that the value of  $\|s\|_k$  for any  $s \notin \mathcal{N}$  has no impact on the algorithm nor on its convergence. Thus we may choose  $\|\cdot\|_k$  as any real function such that it defines a norm on  $\mathcal{N}$  and such that, for all  $j \geq 0$ ,

$$\frac{1}{\kappa_{\text{une}}} \|y\|_{k,j} \leq \|y\| \leq \kappa_{\text{une}} \|y\|_{k,j} \quad \text{for all } y \in \mathcal{N}. \quad (13.10.16)$$

This condition is of course weaker than AN.1, since it only applies in a subspace of  $\mathbb{R}^n$ . We illustrate this by a very useful example. Suppose we are given, at the  $(k, j)$ th iteration, a symmetric matrix  $M_{k,j}$  such that  $N^T M_{k,j} N$  is positive definite and that  $A$  has full rank. Then we may define the dual<sup>224</sup> of the norm  $\|\cdot\|_{k,j}$  by

$$\|v\|_{[k,j]}^2 = \langle v, y \rangle, \quad (13.10.17)$$

---

<sup>224</sup>That this is really the dual norm results from the fact that we define the norm of a vector that is typically the *gradient* of our model.

where  $y$  is the solution of the system

$$\begin{pmatrix} M_{k,j} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} y \\ w \end{pmatrix} = \begin{pmatrix} v \\ 0 \end{pmatrix}, \quad (13.10.18)$$

which is then well defined. If we define  $R$  to be an orthonormal basis of  $\mathcal{N}^\perp$ , we may rewrite this system in the form

$$\begin{pmatrix} N^T M_{k,j} N & N^T M_{k,j} R & N^T A^T \\ R^T M_{k,j} N & R^T M_{k,j} R & R^T A^T \\ AN & AR & 0 \end{pmatrix} \begin{pmatrix} N^T y \\ R^T y \\ w \end{pmatrix} = \begin{pmatrix} N^T v \\ R^T v \\ 0 \end{pmatrix}.$$

The identity  $AN = 0$  and the last line of this system imply that  $ARR^T y = 0$ , which means that  $RR^T y \in \mathcal{N}$  and thus that  $y \in \mathcal{N}$  and  $R^T y = 0$ . The first line of the last system then gives that

$$N^T y = (N^T M_{k,j} N)^{-1} N^T v,$$

and thus, since  $NN^T$  is the orthogonal projector onto  $\mathcal{N}$ , that

$$y = NN^T y = N(N^T M_{k,j} N)^{-1} N^T v. \quad (13.10.19)$$

Substituting this expression in (13.10.17), we obtain that

$$\|v\|_{[k,j]}^2 = \langle N^T v, (N^T M_{k,j} N)^{-1} N^T v \rangle = \|N^T v\|_{(N^T M_{k,j} N)^{-1}}^2,$$

which defines a seminorm,<sup>225</sup> but not a norm, on  $\mathbb{R}^n$ . On the other hand, since we have assumed that  $N^T M_{k,j} N$  is positive definite, it defines a norm on  $\mathcal{N}$ . In particular, if we choose  $M_{k,j} = I$ , we have that  $\|v\|_{k,j} = \|N^T v\|$ , which is the Euclidean norm of the orthogonal projection of  $s$  onto  $\mathcal{N}$ . Condition (13.10.16) then requires that the rightmost eigenvalue of  $N^T M_{k,j} N$  does not tend to infinity (for  $k$  fixed). From Theorem 8.1.6, we then obtain that

$$\|s\|_{k,j} = \|Ns\|_{N^T M_{k,j} N} \text{ and } \|H\|_{\{k,j\}} = \|(N^T M_{k,j} N)^{-\frac{1}{2}} N^T H N (N^T M_{k,j} N)^{-\frac{1}{2}}\|.$$

As for the simpler case where there are no linear equalities, we may relax AN.1 to allow the metric of the (semi)norm to diverge at a controlled speed (see our development of Section 13.7). In the constrained case, and assuming that

$$M_{k,j} \stackrel{\text{def}}{=} H_{k,j} + \mu_k X_{k,j}^{-2}$$

for the pure primal variant of the algorithm or

$$M_{k,j} \stackrel{\text{def}}{=} H_{k,j} + X_{k,j}^{-1} Z_{k,j}$$

for its primal-dual counterpart, we would only require that

$$\lambda_{\min}[N^T M_{k,j} N] \geq \epsilon \text{ and } \|N^T H_{k,j} N\| \leq \kappa_H$$

---

<sup>225</sup>This allows  $\|s\|_{k,j} = 0$  for a nonzero  $s$ , which happens, in this case, for  $0 \neq s \in \mathcal{N}^\perp$ .

for some  $\epsilon > 0$  and  $\kappa_H > 0$ . All that remains is to apply Theorems 13.7.1 and 13.7.2 in the subspace  $\mathcal{N}$ , and all of the good convergence properties of the primal and primal-dual barrier methods immediately follow.

We also see that, if  $g_{k,j}^{\mathcal{N}}$  is computed from the system

$$\begin{pmatrix} M_{k,j} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} g_{k,j}^{\mathcal{N}} \\ w_{k,j} \end{pmatrix} = \begin{pmatrix} g_{k,j} \\ 0 \end{pmatrix},$$

then, applying (13.10.19), we obtain that

$$g_{k,j}^{\mathcal{N}} = N(N^T M_{k,j} N)^{-1} N^T g_{k,j}.$$

The vector  $(N^T M_{k,j} N)^{-\frac{1}{2}} N^T g_{k,j}$  can therefore be seen as the scaled gradient corresponding to the norm induced by  $\|\cdot\|_{k,j}$  on  $\mathcal{N}$ . Most significantly, the scaled gradient is computed quite naturally during the first iteration of the projected preconditioned conjugate gradient method discussed in Section 5.4.1.

If we now turn to the sufficient model decrease condition, we obtain from (13.10.3) that (13.6.4) becomes

$$\begin{aligned} m_{k,j}(x_{k,j}) - m_{k,j}(x_{k,j} + s_{k,j}) \\ \geq \kappa_{\text{sod}} \max \left\{ \|g_{k,j}\|_{[k,j]} \min \left[ \frac{\|g_{k,j}\|_{[k,j]}}{\beta_{\{k,j\}}^{\mathcal{N}}}, \Delta_{k,j}, (1 - \varsigma_k) d_{k,j}^{\text{s}} \right], \right. \\ \left. - \tau_{k,j} \min \left[ \tau_{k,j}^2, \Delta_{k,j}^2, (1 - \varsigma_k)^2 d_{k,j}^{\text{s}} \right] \right\}, \end{aligned}$$

where  $d_{k,j}^{\text{s}} = \text{dist}_{k,j}(x_{k,j}, \partial O^+)$ , where, as indicated by (13.10.4),

$$\beta_{k,j}^{\mathcal{N}} = 1 + \|\nabla_{xx} m_{k,j}(x_{k,j})\|_{\{k,j\}}$$

is assumed to be bounded above (see AM.4h) and where

$$\tau_{k,j} = \lambda_{\min}[N^T \nabla_{xx} m_{k,j}(x_{k,j}) N].$$

Of course, the discussion of this section can be extended to the case where other kinds of inequality constraints are present, exactly as in Sections 13.8 and 13.9. We also observe that the formulation of the problem with bounds and linear equality constraints covers the case where linear inequalities are also included in the problem, at the price of introducing “slack variables”. Indeed, if we consider constraints of the form

$$A_E x = b_E, \quad A_I x \geq b_I,$$

where  $A_I$  is a  $q \times n$  matrix and  $b_I \in \mathbb{R}^q$ , we may reformulate these constraints in the form

$$A_E x = b_E, \quad A_I x - s = b_I, \quad s \geq 0,$$

where  $s \in \mathbb{R}^q$  is the vector of slack variables. The substitutions

$$x \leftarrow \begin{pmatrix} x \\ s \end{pmatrix}, \quad A \leftarrow \begin{pmatrix} A_E & 0 \\ A_I & -I \end{pmatrix}, \quad \text{and} \quad b \leftarrow \begin{pmatrix} b_E \\ b_I \end{pmatrix}$$

then reduce the problem with both equality and inequality constraints to the case where only equality constraints are present. Furthermore, if  $x_0$  is assumed to lie in the interior of the feasible domain, then  $s_0$  may be chosen as  $A_I x_0 - s - b_I > 0$  without altering this property. Finally, the structure of the composite matrix  $A$ , in particular the columns related to the slack variables, may be exploited in the underlying linear algebra.

## Notes and References for Section 13.10

The primal-dual method applied within the affine subspace  $\mathcal{C}_E$  that we have described here is introduced in Conn, Gould, Orban, and Toint (2000). The approach of Byrd, Gilbert, and Nocedal (1996) and Byrd, Hribar, and Nocedal (2000) is similar in spirit, but is more general in that it applies to general equality constraints and does not require the initial point to satisfy these constraints—we shall return to this in Section 15.4.2. An alternative approach is considered by Friedlander, Martínez, and Santos (1994b), where the problem is shown to be equivalent to a larger primal-dual problem (in  $2n + m$  variables) with bound constraints only. This reformulation can then be solved using any of the algorithms discussed at the beginning of the chapter. We also note the algorithm proposed by Lasdon, Plummer, and Yu (1995), which comes in both trust-region and linesearch flavours, and where the step is determined from the unmodified Newton method applied to the barrier problem. Following yet a different line, Gay, Overton, and Wright (1998) and Sargent and Zhang (1998) propose solving the general nonlinear programming problem using the primal-dual model in conjunction with a linesearch and, for the former, a nonmonotone “watchdog” technique. Urban, Tits, and Lawrence (1998) note that sequential quadratic programming algorithms for constrained nonlinear programming can sometimes be recast in the framework of primal-dual methods.

We have not discussed techniques for the “phase 1” of the algorithm discussed in the above section, that is, techniques that ensure one finds a starting point  $x_0 \in \text{ri}\{\mathcal{C}\}$ . In practice, it normally pays to start a significant distance from  $\delta\mathcal{C}$ , as then the Hessian of the barrier function is well behaved. In particular, if  $\mathcal{C}$  is bounded, it often helps to start close to the analytic centre of  $\mathcal{C}$ . For details of such techniques, see, for instance, Zhang (1994) and Gonzalez-Lima, Tapia, and Potra (1998). We also note at this point that the option of choosing  $x_0 \in \text{ri}\{\mathcal{C}\}$  is not the only possibility. Indeed, some algorithms allow their iterates to remain infeasible with respect to the equality constraints, while insisting on (strict) feasibility with respect to the bounds; the linear constraints are then only satisfied at limit points. These methods are typically of the linesearch type and were originally designed for linear or convex problems (see Carpenter et al., 1993, and Simantiraki and Shanno, 1997, for example). An extension of this idea to the nonconvex case has been proposed by Conn, Gould, and Toint (1999), who explicitly penalize infeasibility with respect to the linear constraints at the expense of having to maintain a careful balance between the penalization and penalty parameters. In the context of general inequality constraints, Tseng (1999) discusses a purely primal interior-point trust-region method that allows for an inexact minimization of the logarithmic barrier function but requires the exact solution of the trust-region subproblem.

Another point that we have not discussed is the choice of the starting point  $x_{k,0}$  at the beginning of each inner iteration. While a naive implementation could choose  $x_{k,0} = x_k$ , experience has shown that this is unsatisfactory. Better alternatives relying on extrapolation techniques are proposed by Wright (1995), Conn, Gould, and Toint (1997b), Nash and Sofer (1998), and Gould, Orban, Sartenaer, and Toint (2000a).

Finally, the rate of convergence of the outer primal-dual algorithm is governed by the rate at which the barrier parameter  $\mu_k$  converges to zero. More precisely, we may deduce from the result of Mifflin (1975a) that, in the nondegenerate case and provided each inner iteration is carried out exactly, the R-rate of the sequence  $\{x_k\}$  is the same as the R-rate of  $\{\mu_k\}$  under standard assumptions. Gould, Orban, Sartenaer, and Toint (2000a) prove that the rate of convergence of the sequence of iterates and Lagrange multipliers generated by the algorithm described in this section is componentwise Q-subquadratic in the nondegenerate case for a suitable updating rule for the barrier parameter  $\mu_k$ .

## 13.11 The Affine-Scaling Method

We now turn to another class of well-known “trust-region” methods for the problem of optimizing a (possibly nonconvex) function subject to nonnegativity constraints, where, for simplicity, we shall concentrate on the case of nonnegativities,  $x \geq 0$ . Here, the technique used to prevent the iterates from becoming infeasible is not based on barrier functions, but is still close in spirit to the idea of the first primal inner algorithm: the trust-region radius is adjusted to ensure that the entire trust region lies within the feasible domain. The origin of the method that we now discuss, called the *affine-scaling method*, is in the field of linear programming, and the main idea is simply to minimize a quadratic model of the objective function of the form

$$m_k(x_k + s) = m_k(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle, \quad (13.11.1)$$

where  $H_k$  is an  $n \times n$  symmetric matrix, in a trust region whose shape is similar, but not identical, to that of the scaled trust regions of Section 13.7. In this case, we choose to use the scaled norm defined by

$$\|s\|_k^2 = \|s\|_{X_k^{-2}}^2 = \|X_k^{-1}s\|^2 = \langle s, X_k^{-2}s \rangle,$$

and, when  $\Delta_k \leq 1$ , the corresponding trust region  $\mathcal{B}_k$  is then referred to as *Dikin’s ellipsoid*. The resulting algorithm may be described as follows.

**Algorithm 13.11.1: Affine-scaling algorithm for bounds**

**Step 0: Initialization.** An initial point  $x_0 \in \text{ri}\{\mathcal{C}\}$  and an initial trust-region radius  $\Delta_0 = \Delta_{\max} \in (0, 1)$  are given. The constants  $\eta_1, \eta_2, \gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Compute  $f(x_0)$  and its gradient, and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k^2 = \langle \cdot, X_k^{-2} \cdot \rangle$  and define the model  $m_k$  as in (13.11.1).

**Step 2: Step calculation.** Compute a step  $s_k$  such that  $x_k + s_k \in \mathcal{B}_k$  such that it sufficiently reduces the model  $m_k$  in the sense that

$$\begin{aligned} m_k(x_k) - m_k(x_k + s_k) \\ \geq \kappa_{\text{sod}} \max \left\{ \|g_k\|_{[k]} \min \left[ \frac{\|g_k\|_{[k]}}{\beta_{\{k\}}}, \Delta_k \right], -\tau_k \min \left[ \tau_k^2, \Delta_k^2 \right] \right\}. \end{aligned} \quad (13.11.2)$$

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Then, if  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \Delta_{\max}] & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

We immediately notice that, besides the change in scaling, the affine-scaling algorithm imposes a maximum trust-region radius  $\Delta_{\max} < 1$ . As a consequence, since this ensures that

$$\|X_k^{-1} s_k\| < 1,$$

we always have that  $x_k + s_k \in \text{ri}\{\mathcal{C}\}$ , and there is no longer any need to check feasibility. This of course simplifies the algorithm, but also may impose short steps, as we already discussed. The choice of the shape of the trust region is crucial in our formulation of the affine-scaling algorithm in that it immediately allows us to deduce the following useful property.

**Theorem 13.11.1** Suppose that AF.1–AF.3, AM.1–AM.3, and AM.4i hold. Suppose furthermore that the sequence of iterates  $\{x_k\}$  generated by the affine-scaling algorithm remains in a bounded subset of the positive orthant. Then

$$\lim_{k \rightarrow \infty} \|X_k g_k\| = 0. \quad (13.11.3)$$

**Proof.** We wish to apply the convergence results expressed in dual norms, and, in particular, use the conclusions of Section 8.1.4, which is (13.11.3). We therefore have

to verify that the relevant assumptions, that is, AN.1b, AN.3, AM.4b (restricted to  $\text{ri}\{\mathcal{C}\}$ ), and AA.1b, hold.

We first define  $\mathcal{D}$  to be a compact subset of the positive orthant containing all the iterates of the sequence  $\{x_k\}$ , and let

$$\kappa_{\text{ubx}} = \max_{x \in \mathcal{D}} \|x\|_\infty.$$

Then it is easy to verify that

$$\|X_k^{-1}v\| \geq \frac{1}{\kappa_{\text{ubx}}} \|v\|$$

for each  $v \in \mathbb{R}^n$ , and AN.1b thus holds. Furthermore, the norm  $\|\cdot\|_{X_k^{-2}}$  is continuous on  $\text{ri}\{\mathcal{C}\}$  by construction, and we also know that  $g(x) = \nabla_x f(x)$  is continuous because of AF.1 and AM.3. Hence  $\|g(x)\|_{X_k^{-2}}$  is continuous on  $\text{ri}\{\mathcal{C}\}$ , and AN.3 follows. Now, because AN.1b holds, Theorem 8.1.7 (p. 261) ensures that AM.4b also holds on  $\mathcal{C}$ , because we have assumed AM.4i. Finally, (13.11.2) is nothing but AA.1b for the first-order measure defined by the dual norm of the gradient

$$\|g_k\|_{[k]} = \|X_k g_k\|,$$

where we have used Lemma 8.1.6 (p. 260) to deduce the form of the dual norm. Hence the conclusions of Section 8.1.4 apply, and we deduce that (13.11.3) holds.  $\square$

However, as we now show, this result is not sufficient, in general, to enforce convergence to first-order critical points. Consider the two-dimensional problem

$$\min_{x \geq 0} \frac{9}{10}[x]_1 - \frac{1}{10}[x]_2.$$

This problem has a solution at infinity,<sup>226</sup> and yet the affine-scaling algorithm using the best possible model, that is,  $m_k(x_k + s) = f(x_k + s)$ , may produce a sequence of iterates converging to the origin, where complementarity holds, but which is not first-order critical. Suppose that  $x_0 = \alpha_0(1, 1)^T$  for some  $\alpha_0 > 0$ . This implies that the initial trust region is circular and of radius  $\alpha_0 \Delta_{\max}$ . Defining

$$g = \begin{pmatrix} \frac{9}{10} \\ -\frac{1}{10} \end{pmatrix},$$

we also suppose that

$$\Delta_{\max} = \|g\| = \sqrt{\frac{41}{50}} < 1.$$

---

<sup>226</sup>It is of course possible to transform the problem to make it finite by adding upper bounds on the variables.

We then obtain that the Cauchy point at iteration 0 is given by

$$x_0^C = \alpha_0 \begin{pmatrix} 1 - \frac{9}{10} \frac{\Delta_{\max}}{\|g\|} \\ 1 + \frac{1}{10} \frac{\Delta_{\max}}{\|g\|} \end{pmatrix} = \begin{pmatrix} \frac{1}{10} \\ \frac{11}{10} \end{pmatrix}.$$

If we are satisfied by a model reduction that is, say, half of that obtained at the Cauchy point, then any point on the intersection of the trust region and the line orthogonal to the negative gradient at  $\frac{1}{2}(x_0 + x_0^C)$  ensures that AA.1b holds. A little algebra shows that this segment intersects the line  $[x]_1 = [x]_2$  at the point  $\frac{1}{2}x_0$ , which we then select as trial point  $x_0 + s_0$ . Since the objective function is linear and we suppose AM.1–AM.3, we deduce that  $\rho_0 = 1$ , and thus  $x_1 = x_0 + s_0$  and  $\Delta_1 = \Delta_{\max}$ . Thus all conditions that were satisfied at  $x_0$  remain satisfied at  $x_1$ , and we conclude that

$$x_k = (\frac{1}{2})^k \begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

which converges to the origin, as announced. This iteration is illustrated in Figure 13.11.1.

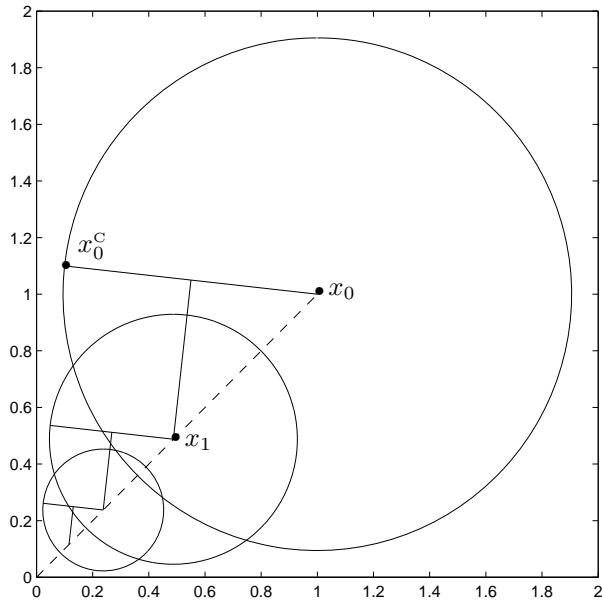


Figure 13.11.1: Three iterations of a sequence generated by the affine-scaling algorithm and converging to a noncritical point starting from  $x_0 = (1, 0^T)$ .

The problem is that the steps become asymptotically too short in the second variable, despite the fact that the direction of the negative gradient should encourage large steps in this direction.

Of course, this counterexample depends on the fact that we have allowed for the inexact solution of the trust-region problem, by accepting a trial point at which the model reduction is only half of that at the Cauchy point. What would happen if we were more strict? Returning to our example, we see that if we choose  $x_k + s_k = x_k^C = x_k^M$ , then

$$[x_{k+1}]_2 \geq [x_k]_2, \quad (13.11.4)$$

and convergence to the origin cannot occur. We could even allow some inaccuracy in choosing  $x_k + s_k$  in a small neighbourhood of  $x_k^M$  without altering this property. This discussion shows that the accuracy with which the trust-region problem must be solved, if one is to obtain convergence to first-order critical points,<sup>227</sup> varies with the value of the gradient components at (unknown) noncritical points where complementarity holds. This is extremely inconvenient in practice, as it rules out computing the step using, for example, the Steihaug–Toint method. One should, however, notice that our counterexample does not apply if  $s_k$  is computed as the exact minimizer of the model in a trust region whose radius is not exactly  $\Delta_k$ , as this again enforces (13.11.4) for any value of  $\Delta_k < 1$ . This can be achieved if one uses Algorithm 7.3.4 *in exact arithmetic*. These potential difficulties with the affine-scaling algorithm lead us to investigate a variant of the idea where they do not occur, which is the object of the next section.

## Notes and References for Section 13.11

The idea discussed here is similar but not identical to that of Dikin’s ellipsoid introduced, in the context of affine-scaling methods, by Dikin (1967) and Dikin and Zorkaltsev (1980), and subsequently rediscovered by Ye (1989) and Ye and Tse (1989). Affine-scaling methods have enjoyed a lot of attention for the case where the objective function is linear or quadratic, most often convex, or concave. Among the vast literature on this subject, we cite Gonzaga (1991), Ye (1992), Han, Pardalos, and Ye (1992), Sun (1993), Tsuchiya (1993), Bonnans and Pola (1997), Bonnans and Bouhtou (1995), Monteiro and Tsuchiya (1998), and Coleman and Liu (1999). The affine-scaling algorithm for convex or concave programs has been considered by Monteiro and Wang (1998), where trust-region methodology is used in a form very close to that of our Chapter 6. They allow for an inexact solution of the trust-region subproblem, in that the step  $s_k$  is the exact minimizer of the model in an ellipsoid of radius  $\bar{\Delta}_k$  that is close (in relative terms) to  $\Delta_k$ .

## 13.12 The Method of Coleman and Li

### 13.12.1 The Algorithm

As we have seen, ignoring the gradient of the objective function at the current iterate in the definition of the local norm may not be without drawbacks. We thus devote this section to the discussion of an algorithm proposed by Coleman and Li that can

---

<sup>227</sup>Observe that this is merely a *necessary* condition for optimality.

be viewed as an attempt to avoid this problem by redefining the norm  $\|\cdot\|_k$  and by allowing trust-region radii that exceed 1, therefore letting it extend into the infeasible region, as we did in Sections 13.2–13.7.

As the algorithm is intended for problems with bound constraints, we continue to assume that  $\mathcal{C}$  is the positive orthant of  $\mathbb{R}^n$ . The main argument is then to reformulate the first-order criticality conditions for problem (13.4.1) by introducing a well-chosen scaling matrix. For any  $x \in \text{ri}\{\mathcal{C}\}$ , define the vector  $v(x) \in \mathbb{R}^n$  as follows:

$$[v(x)]_i = \begin{cases} [x]_i & \text{if } [\nabla_x f(x)]_i \geq 0, \\ -1 & \text{if } [\nabla_x f(x)]_i < 0 \end{cases} \quad (13.12.1)$$

for  $i = 1, \dots, n$ . We then define the diagonal scaling matrix (see (6.7.1) [p. 164]) by

$$S(x) \stackrel{\text{def}}{=} \text{diag}\left(|[v(x)]_1|^{\frac{1}{2}}, \dots, |[v(x)]_n|^{\frac{1}{2}}\right). \quad (13.12.2)$$

The motivation for these definitions is contained in the following equivalence.

**Theorem 13.12.1** Suppose that AF.1 holds. Then  $x_* \geq 0$  is a first-order critical point of problem (13.4.1) if and only if

$$S(x_*)^2 \nabla_x f(x_*) = S(x_*) \nabla_x f(x_*) = 0. \quad (13.12.3)$$

**Proof.** Let  $\mathcal{A}$  contain the indices of the active set at  $x_*$ , that is,

$$\mathcal{A} = \{i \in \{1, \dots, n\} \mid [x_*]_i = 0\}.$$

We first show that (13.12.3) implies criticality. If  $i \in \mathcal{A}$ , then the  $i$ th equation of (13.12.1) and (13.12.3) gives that  $[\nabla_x f(x_*)]_i \geq 0$ . Otherwise, (13.12.3) implies that  $[\nabla_x f(x_*)]_i = 0$ , as required.

Conversely, if  $i \notin \mathcal{A}$ , the first-order criticality of  $x_*$  implies that  $[\nabla_x f(x_*)]_i = 0$ , and the  $i$ th equation of (13.12.3) holds. On the other hand, if  $i \in \mathcal{A}$  and  $[\nabla_x f(x_*)]_i \geq 0$ , (13.12.1) yields that  $[v(x_*)]_i = [x_*]_i = 0$ , and the  $i$ th equation of (13.12.3) is also satisfied.  $\square$

In order to find a first-order critical point we may thus consider solving the system of nonlinear equations

$$S(x)^2 \nabla_x f(x) = 0. \quad (13.12.4)$$

Observe that, despite the discontinuity of  $v(x)$  as a function of  $x$ , the left-hand side of this system is continuous. It is, however, not everywhere differentiable, as discontinuities in the gradient occur for points on the boundary of the feasible region. These points are avoided by requiring, as we have throughout this chapter, that all iterates remain strictly interior (to the positive orthant). Observe also that, although  $v(x)$  is

potentially discontinuous at points at which one of the components of the gradient vanishes,  $S(x)^2 \nabla_x f(x)$  is nevertheless continuous at such points.

So consider applying Newton's method to (13.12.4) at a point  $x_k > 0$  for which  $S(x)^2 \nabla_x f(x)$  is differentiable. The corresponding step  $s_k$  is then obtained from the system

$$\left[ S(x_k)^2 \nabla_{xx} f(x_k) + \text{diag}\left(\left[\nabla_x f(x_k)\right]_1, \dots, \left[\nabla_x f(x_k)\right]_n\right) J_k \right] s_k = -S(x_k)^2 \nabla_x f(x_k),$$

where  $J_k = J(x_k)$  is the Jacobian matrix of the vector  $|v(x)|^{\frac{1}{2}}$  at  $x_k$  when it is differentiable. If we now replace the objective function by a quadratic model  $m_k$  of the form

$$m_k^f(x_k + s) = f(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, H_k s \rangle \quad (13.12.5)$$

satisfying AM.1–AM.4 and use the notation  $S_k = S(x_k)$ ,  $G_k = \text{diag}([g_k]_1, \dots, [g_k]_n)$ , we may write a model of this system in the more compact form

$$\left( S_k^2 H_k + G_k J_k \right) s_k = -S_k^2 g_k.$$

Observe that  $J_k$  is diagonal, and

$$[J_k]_{ii} = \begin{cases} 1 & \text{if } [g_k]_i > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Hence we have that

$$G_k J_k = \text{diag}\left(\max\left[[g_k]_1, 0\right], \dots, \max\left[[g_k]_n, 0\right]\right) \stackrel{\text{def}}{=} \bar{G}_k.$$

The system defining the (quasi-)Newton step thus becomes

$$\left( S_k^2 H_k + \bar{G}_k \right) s_k = -S_k^2 g_k.$$

Multiplying this equation on the left by  $S_k^{-1}$  and introducing the new variable  $w_k = S_k w_k$ , we may finally write the equation of the Newton step in the form

$$\left( S_k H_k S_k + \bar{G}_k \right) w_k = -S_k g_k, \quad (13.12.6)$$

where we have used the identity  $S_k^{-1} \bar{G}_k S_k = \bar{G}_k$ . We now note that (13.12.6) is the Newton's equation for the quadratic minimization problem

$$\min_s m^s(x_k + s),$$

where

$$\begin{aligned} m_k^s(x_k + w) &= f(x_k) + \langle S_k g_k, w \rangle + \frac{1}{2} \left\langle w, S_k \left( H_k + M_k \right) S_k w \right\rangle \\ &\stackrel{\text{def}}{=} f(x_k) + \langle g_k^s, w \rangle + \frac{1}{2} \left\langle w, \left( H_k^s + M_k^s \right) w \right\rangle. \end{aligned} \quad (13.12.7)$$

In this definition, we have used the superscript  $s$  to indicate scaled quantities, as we did in Section 6.7, and we have set

$$M_k = S_k^{-2} \bar{G}_k = |V_k|^{-1} \bar{G}_k,$$

with  $|V_k| \stackrel{\text{def}}{=} \text{diag}(|[v(x_k)]_1|, \dots, |[v(x_k)]_n|)$ . Note that  $M_k^s = \bar{G}_k$ .

Furthermore, we have that a minimizer of  $m_k^s$  is strongly related to the solution of our original problem.

**Theorem 13.12.2** Suppose that AF.1–AF.3 hold and that  $x_* \geq 0$ . Then  $g_*^s = 0$  if and only if  $x_*$  is a first-order critical point of problem (13.4.1). Moreover,  $[\nabla_{xx}f(x_*)]^s + M_*^s$  is positive semidefinite and  $g_*^s = 0$  if and only if  $x_*$  is a weak second-order critical point.

**Proof.** The first part of the statement follows from Theorem 13.12.1. Suppose now that  $x_*$  is a weak second-order critical point. Then it is also a first-order one and we deduce from the first part that  $g_*^s = 0$ . Define  $\mathcal{N}$  to be the null-space of the space spanned by the gradients of the active constraints, that is,

$$\mathcal{N} = \{u \in \mathbb{R}^n \mid [u]_i = 0 \text{ whenever } [x_*]_i = 0\}.$$

Consider first a vector  $u \in \mathcal{N}$ . Since  $x_*$  is first-order critical, we have that  $\langle u, g_* \rangle = 0$  and that

$$\langle u, \bar{G}_* u \rangle = \langle u, M_*^s u \rangle = 0.$$

Furthermore, we also have that  $S_* u \in \mathcal{N}$  from (13.12.1). Thus we have that

$$\langle u, [\nabla_{xx}f(x_*) + M_*]^s u \rangle = \langle S_* u, [\nabla_{xx}f(x_*)]S_* u \rangle = \langle u', [\nabla_{xx}f(x_*)]u' \rangle$$

for  $u' = S_* u \in \mathcal{N}$ . If we now consider  $u \in \mathcal{N}^\perp$ , then (13.12.1) implies that  $S_* u = 0$  and therefore that

$$\langle u, [\nabla_{xx}f(x_*) + M_*]^s u \rangle = \langle S_* u, [\nabla_{xx}f(x_*) + M_*]S_* u \rangle = 0.$$

The theorem then immediately follows from the definition of the second-order necessary optimality conditions.  $\square$

It is thus relatively natural to consider a step  $s_k = S_k w_k$ , where  $w_k$  is chosen to reduce  $m_k^s(x_k + w)$ , so long as we believe that the model  $m_k^f$  (a vital constituent of  $m_k^s$ ) correctly approximates the objective function. If we believe this to be the case within a trust region around  $x_k$ , as we have done so often in the past, we may thus compute  $w_k$  as an (approximate) minimizer of  $m_k^s$  in a trust region of the form

$$\mathcal{B}_k^s = \{w \in \mathbb{R}^n \mid \|w\| \leq \Delta_k\}.$$

As we saw in Section 6.7, this is equivalent to (approximately) minimizing the model

$$m_k(x_k + s) = f(x_k) + \langle g_k, s \rangle + \frac{1}{2} \langle s, [H_k + M_k]s \rangle \quad (13.12.8)$$

in a scaled trust region of the form

$$\mathcal{B}_k = \{s \in \mathbb{R}^n \mid \|S_k^{-1}s\| \leq \Delta_k\}.$$

In other words, the (approximate) minimization of the model (13.12.8) is performed in a trust region defined by the ellipsoidal norm

$$\|s\|_k^2 = \langle s, S_k^{-2}s \rangle = \|s\|_{S_k^{-2}}.$$

We thus obtain Algorithm 13.12.1.

**Algorithm 13.12.1: Coleman and Li's algorithm (version 1)**

**Step 0: Initialization.** An initial point  $x_0 \in \text{ri}\{\mathcal{C}\}$  and an initial trust-region radius  $\Delta_0$  are given. The constants  $\eta_1$ ,  $\eta_2$ ,  $\gamma_1$ , and  $\gamma_2$  are also given and satisfy the conditions (6.1.3) (p. 116). Constant  $\varsigma_{\max} \in (0, 1)$  and  $\epsilon > 0$  are also given. Compute  $f(x_0)$  and its gradient, and set  $k = 0$ .

**Step 1: Model definition.** Choose  $\|\cdot\|_k^2 = \langle \cdot, S_k^{-2} \cdot \rangle$  and define the model  $m_k$  as in (13.12.8).

**Step 2: Step calculation.** Choose a  $\varsigma_k \in (0, \varsigma_{\max}]$  and define  $d_k = \text{dist}(x_k, \partial\mathcal{C})$ . Compute a step  $s_k$  such that  $x_k + s_k \in \mathcal{B}_k$  and

$$\text{dist}(x_k, +s_k, \partial\mathcal{C}) \geq \varsigma_k d_k \quad (13.12.9)$$

and such that it sufficiently reduces the model  $m_k$  in the sense that AA.1b and AA.2b hold for the first-order criticality measure  $\pi_k = \min[\|g_k^S\|, \epsilon]$ .

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}. \quad (13.12.10)$$

Then if  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

It is also interesting to point out, at this stage, the similarity between the term  $M_k = S_k^{-2}\bar{G}_k = |V_k|^{-1}\bar{G}_k$  in the Hessian of the model  $m_k(x_k + s_k)$ , the term  $\mu_k X_k^{-2}$ , which appears in the model of the primal log-barrier methods, and the term  $X_k^{-1}Z_k$ , which appears in that of the primal-dual method. Indeed, all three terms are of the form

$$\text{diag} \left( \frac{\text{estimate of the dual variables}}{\text{distance to the bound}} \right),$$

because we know, from the first-order optimality conditions and AA.8, that  $\mu_k X_k^{-1}e$ ,  $z_k$ , and  $|g_k|$  converge to the optimal dual variables at  $x_*$ . Asymptotically, these three models therefore share the same second derivative matrix (assuming that  $\nabla_{xx}m_k^f(x_k)$  converges to  $\nabla_{xx}f(x_*)$ ). They differ, however, in their gradients: the gradients of the models at  $x_k$  are those of the log-barrier function,  $\nabla_x f(x_k) - \mu_k X_k^{-1}e$ , for the primal and primal-dual algorithms, while  $\nabla_x m_k(x_k)$  is equal to  $\nabla_x f(x_k)$  for the Coleman–Li model. This observation thus leads to the question of which function  $m_k(x_k + s_k)$  is actually a model of and shows that we should consider it as a model of the objective function  $f(x)$  if we wish to ensure that it is adequate to first order.

At variance with the primal and primal-dual algorithm, the method of Coleman and Li does not involve any outer iteration to drive a barrier parameter to zero, which is an advantage. Note that we had to reintroduce the “sufficient feasibility” constraint (13.12.9) because we no longer assume that the trust region remains included in the feasible domain. However, we haven’t reintroduced the terms involving the distance to the boundary of the feasible region in the model decrease condition since we are simply supposing that AA.1b and AA.2b hold. Is this consistent? Fortunately yes, as we now show successively for AA.1b and AA.2b.

### 13.12.2 Convergence Theory

To ensure a sufficient (first-order) model decrease, the idea is to use the scaled Cauchy point as defined in Section 8.1.6, that is, to use the steepest-descent direction in the scaled norm. Ideally, we could thus apply Theorem 8.1.12 (p. 270) and deduce immediately that AA.1b holds. However, while this implication is true for the unconstrained problem, the presence of the constraints must be taken into account in our present context. We therefore combine the ideas of the scaled and constrained Cauchy points of Section 13.3.

**Theorem 13.12.3** If the model is of the form (13.12.5) and if we define the *constrained scaled Cauchy point*  $x_k^{\text{CSC}}$  as the solution of the minimization problem

$$\min_{t \geq 0} m_k(x_k - tS_k^2 g_k)$$

subject to the constraints

$$t\|S_k^2 g_k\|_k \leq \Delta_k \quad \text{and} \quad \text{dist}(x_k - tS_k^2 g_k, \partial\mathcal{C}) \geq \varsigma_k \text{dist}(x_k, \partial\mathcal{C}), \quad (13.12.11)$$

then, if  $\|g_k\| \leq \kappa_{\text{ubg}}$  for some  $\kappa_{\text{ubg}} > 0$ , there exists a constant<sup>228</sup>  $\kappa_{\text{msd}} \in (0, 1]$  such that

$$m_k(x_k) - m_k(x_k^{\text{CSC}}) \geq \frac{1}{2}\pi_k \min \left[ \frac{\pi_k}{\beta_{\{k\}}}, \Delta_k \right],$$

where

$$\pi_k = \min \left[ \|g_k^{\text{s}}\|, \kappa_{\text{msd}} \right]$$

and where  $\beta_{\{k\}}$  is defined in (8.1.22) (p. 262).

**Proof.** The reader might wish to compare this proof with that of Theorems 8.1.12 (p. 270) and 13.3.1. Indeed, the current result may be viewed as the formal statement of a scaled version of the latter.

We start by considering the maximal steplength along the scaled steepest-descent direction that preserves sufficient feasibility in the sense of the second part of (13.12.11). This step is given by

$$t_k^{\text{F}} = \min_{i|[g_k]_i > 0} \frac{(1 - \varsigma_k)[x_k]_i}{[S_k^2 g_k]_i}.$$

Now (13.12.1) and (13.12.2) give that

$$[S_k^2 g_k]_i = [x_k]_i [g_k]_i \leq \kappa_{\text{ubg}} [x_k]_i$$

for all  $i$  such that  $[g_k]_i > 0$ , which in turn yields that

$$t_k^{\text{F}} \geq \frac{1 - \varsigma_k}{\kappa_{\text{ubg}}} \geq \frac{1 - \varsigma_{\max}}{\kappa_{\text{ubg}}} \stackrel{\text{def}}{=} \kappa_{\text{msd}} > 0.$$

Without loss of generality, we assume that  $\kappa_{\text{msd}} \leq 1$ . We also note that, for all  $t \geq 0$ ,

$$m_k(x_k - t S_k^2 g_k) = m_k(x_k) - t \|g_k^{\text{s}}\|^2 + \frac{1}{2} t^2 \langle g_k^{\text{s}}, H_k^{\text{s}} g_k^{\text{s}} \rangle, \quad (13.12.12)$$

where we have used the identities  $g_k^{\text{s}} = S_k g_k$  and  $H_k^{\text{s}} = S_k H_k S_k$ . We now consider the case where the curvature of the scaled model along the scaled steepest descent is positive, that is, when

$$\langle g_k^{\text{s}}, H_k^{\text{s}} g_k^{\text{s}} \rangle > 0, \quad (13.12.13)$$

and compute the value of the parameter  $t$  at which the unique minimum of (13.12.12) is attained. Let us denote this optimal parameter by  $t_k^*$ . Differentiating (13.12.12) with respect to  $t$  and equating the result to zero, we obtain that

$$t_k^* = \frac{\|g_k^{\text{s}}\|^2}{\langle g_k^{\text{s}}, H_k^{\text{s}} g_k^{\text{s}} \rangle}. \quad (13.12.14)$$

Two subcases may then occur. The first is when

$$t_k^* \|g_k^{\text{s}}\| \leq \min[\Delta_k, \kappa_{\text{msd}}].$$

<sup>228</sup>“msd” stands for “minimum step along the negative scaled gradient”.

Then  $t_k^{\text{CSC}} = t_k^*$  and we may replace this expression in the model decrease (13.12.12), which allows us to deduce that

$$m_k(x_k) - m_k(x_k^{\text{CSC}}) \geq \frac{\|g_k^s\|^4}{\langle g_k^s, H_k^s g_k^s \rangle} - \frac{1}{2} \frac{\|g_k^s\|^4}{\langle g_k^s, H_k^s g_k^s \rangle} \geq \frac{\|g_k^s\|^2}{2\beta_{\{k\}}}. \quad (13.12.15)$$

Here we used the fact that  $|\langle g_k^s, H_k^s g_k^s \rangle| \leq \|g_k^s\|^2 \beta_{\{k\}}$  because of the Cauchy–Schwarz inequality and the definition of  $\beta_{\{k\}}$ . The second case occurs when

$$t_k^* \|g_k^s\| > \min[\Delta_k, \kappa_{\text{msd}}]. \quad (13.12.16)$$

Then the line minimum is outside the trust region or at a distance exceeding  $\kappa_{\text{msd}}$ , and we have that

$$t_k^{\text{CSC}} \|g_k^s\| = \min[\Delta_k, \kappa_{\text{msd}}]. \quad (13.12.17)$$

Combining (13.12.14), (13.12.16), and (13.12.17), we see that

$$\langle g_k^s, H_k^s g_k^s \rangle \leq \frac{\|g_k^s\|^2}{t_k^{\text{CSC}}}.$$

Substituting this last inequality in (13.12.12) and using (13.12.17), we obtain that

$$\begin{aligned} m_k(x_k) - m_k(x_k^{\text{CSC}}) &= t_k^{\text{CSC}} \|g_k^s\|^2 - \frac{1}{2} [t_k^{\text{CSC}}]^2 \langle g_k^s, H_k^s g_k^s \rangle \\ &\geq t_k^{\text{CSC}} \|g_k^s\|^2 - \frac{1}{2} t_k^{\text{CSC}} \|g_k^s\|^2 \\ &= \frac{1}{2} \|g_k^s\| \min[\Delta_k, \kappa_{\text{msd}}]. \end{aligned} \quad (13.12.18)$$

Finally, we consider the case where the curvature of the model along the steepest-descent direction is negative, that is, when (13.12.13) is violated. We then obtain from (13.12.12) that

$$m_k(x_k - tg_k) = m_k(x_k) - t\|g_k^s\|^2 + \frac{1}{2} t^2 \langle g_k^s, H_k^s g_k^s \rangle \leq m_k(x_k) - t\|g_k\|^2 \quad (13.12.19)$$

for all  $t \geq 0$ . In that case, it is easy to see that (13.12.17) holds. Combining this equality and (13.12.19), we deduce that

$$m_k(x_k) - m_k(x_k^C) \geq \|g_k^s\| \min[\Delta_k, \kappa_{\text{msd}}] \geq \frac{1}{2} \|g_k^s\| \min[\Delta_k, \kappa_{\text{msd}}]. \quad (13.12.20)$$

We may then conclude that (13.12.15), (13.12.18), and (13.12.20) imply that

$$m_k(x_k) - m_k(x_k^C) \geq \frac{1}{2} \|g_k^s\| \min \left[ \frac{\|g_k^s\|}{\beta_{\{k\}}}, \Delta_k, \kappa_{\text{msd}} \right] \geq \frac{1}{2} \pi_k \min \left[ \frac{\pi_k}{\beta_{\{k\}}}, \Delta_k \right],$$

where we have used the definition of  $\pi_k$  and the fact that both  $\beta_{\{k\}}$  and  $\kappa_{\text{msd}}$  are at most 1 to derive the last inequality. This completes the proof.  $\square$

We therefore see that it is again reasonable to assume AA.1b in Step 2 of the algorithm provided we can ensure that all gradients remain bounded, which is, for instance, the case when the iterates stay in a bounded domain. We now examine the convergence to first-order critical points.

**Theorem 13.12.4** Suppose that AF.1–AF.3 hold for the objective function. Suppose also that its model  $m_k^f$  satisfies AM.1–AM.4. Suppose finally that the iterates generated by the Coleman–Li algorithm remain in a bounded domain of the positive orthant. Then we obtain that

$$\lim_{k \rightarrow \infty} \|g_k^s\| = 0,$$

and therefore every limit point  $x_*$  of the sequence of iterates is first-order critical.

**Proof.** The key is then to show that we may apply the conclusions of Section 8.1.2 (p. 251). We thus have to verify first that the complete model  $m_k$  satisfies AM.1–AM.3, AM.4b,<sup>229</sup> and AA.1b, and that the norm  $\|\cdot\|_k$  satisfies AN.1b.

We first observe that the first three assumptions must hold for  $m_k$  since they are satisfied for  $m_k^f$ . We next note that the bounded nature of the sequence  $\{x_k\}$  implies that

$$[x_k]_i \leq \kappa_{\text{ubx}} \quad (i = 1, \dots, n) \quad (13.12.21)$$

for all  $k \geq 0$  and for some  $\kappa_{\text{ubx}}$  which we assume, without loss of generality, to be at least equal to 1. Thus  $[v(x_k)]_i$  is uniformly bounded with respect to  $k$  and for all  $i = 1, \dots, n$  by  $\sqrt{\kappa_{\text{ubx}}}$  and, consequently,  $\|V_k^{-\frac{1}{2}}\| \geq \kappa_{\text{ubx}}^{-\frac{1}{2}}$ . Remembering now that  $\|s\|_k = \|S_k^{-1}s\| = \|V_k^{-\frac{1}{2}}s\|$ , we deduce that

$$\|s\|_k \geq \frac{1}{\sqrt{\kappa_{\text{ubx}}}} \|s\|,$$

which shows that AN.1b is satisfied. We now verify that AM.4b holds, that is, that the Hessian of the scaled model is uniformly bounded. Note that this Hessian is given by

$$\nabla_{xx} m_k^s(x_k + s) = S_k(H_k + M_k)S_k = S_k H_k S_k + \bar{G}_k.$$

Furthermore, the boundedness of the sequence  $\{x_k\}$  and assumptions AF.1 and AM.3 imply that there exists  $\kappa_{\text{ubg}} > 0$  such that

$$\|g_k\| \leq \kappa_{\text{ubg}}. \quad (13.12.22)$$

Hence, we obtain from (8.1.16) (p. 260), AM.4 for  $m_k^f$ , (13.12.21), and (13.12.22) that

$$\|\nabla_{xx} m_k(x_k + s)\|_{\{k\}} = \|\nabla_{xx} m_k^s(x_k + s)\| \leq \kappa_{\text{ubx}} \kappa_{\text{umh}} + \kappa_{\text{ubg}},$$

from which we may then deduce that AM.4b holds. Finally, we recall that the algorithm ensures that AA.1b holds (on the basis of Theorem 13.12.3) if we prove that  $\pi_k$  is a proper first-order criticality measure. While it is certainly a nonnegative real

---

<sup>229</sup>We could consider an extension of the results of Section 8.1.2 that would only require AM.4b to hold in  $\text{ri}\{\mathcal{C}\}$ , and consequently assume AM.4i instead of AM.4. We do not consider this generalization in order to simplify the presentation.

function of  $x_k$ , we still have to verify that it is “continuous across iterations”, that is, that (8.1.2) (p. 250) holds. But it is clear that the function  $\min[\|S(x)g(x)\|, \kappa_{\text{msd}}]$  is continuous in  $x$ , which implies that

$$\left| \min \left[ \|g_k^S\|, \kappa_{\text{msd}} \right] - \min \left[ \|g_t^S\|, \kappa_{\text{msd}} \right] \right| \rightarrow 0$$

when  $\|x_k - x_t\|$  tends to zero, and (8.1.2) holds. As a consequence of this discussion, we may deduce from (8.1.9) (p. 254) that

$$\lim_{k \rightarrow \infty} \min \left[ \|g_k^S\|, \kappa_{\text{msd}} \right] = 0$$

and thus that

$$\lim_{k \rightarrow \infty} \|S_k g_k\| = \lim_{k \rightarrow \infty} \|g_k^S\| = 0.$$

The last statement of the theorem then follows from the first part of Theorem 13.12.2 and the fact that the bounded nature of the sequence  $\{x_k\}$  implies the existence of at least one limit point.  $\square$

We have thus established that Coleman and Li’s method converges to constrained<sup>230</sup> first-order critical points. However, convergence to second-order critical points does not follow directly from the discussion of Section 8.1.4 since the latter requires additionally that AM.5b, AM.6b, and AA.2b hold. While AM.6b is of no concern as  $m_k$  is quadratic, this is not the case with AM.5b and AA.2b.

We examine the case of AA.2b. Again, the difficulty is that the eigenpoint as described in Chapters 6 and 8 may be infeasible, and therefore that the model reduction obtained at this point may not be achievable for a feasible point. However, as was the case for the constrained scaled Cauchy point, we may define a *scaled constrained eigenpoint* and show that the model reduction obtained at this point ensures AA.2b. More precisely, let  $z_k$  be the unit eigenvector corresponding to the leftmost eigenvalue of  $H_k^S + M_k^S$  whose sign is chosen to ensure that  $\langle g_k, w_k \rangle \leq 0$ .

**Theorem 13.12.5** Suppose that AF.1–AF.3 hold for the objective function. Suppose also that its model  $m_k^f$  satisfies AM.1–AM.4. Suppose also that the iterates generated by the Coleman–Li algorithm remain in a bounded domain of the positive orthant. Suppose finally that  $\tau_k < 0$  and that the constrained scaled eigenpoint  $x_k^{\text{CSE}}$  is defined to be the solution of the minimization problem

$$\min_{t \geq 0} m_k(x_k - tS_k z_k),$$

subject to the constraints

$$t\|S_k z_k\|_k \leq \Delta_k \text{ and } \text{dist}(x_k - tS_k z_k, \partial\mathcal{C}) \geq \varsigma_k \text{dist}(x_k, \partial\mathcal{C}). \quad (13.12.23)$$

---

<sup>230</sup>Observe that, in general,  $\|g_k\| \neq 0$ , despite the fact that  $\lim_{k \rightarrow \infty} \|g_k^S\| = 0$ .

Then there exists a constant  $\kappa_{\text{sod}} \in (0, 1]$  such that

$$m_k(x_k) - m_k(x_k^{\text{CSC}}) \geq -\kappa_{\text{sod}} |\tau_k| \min \left[ \tau_k^2, \Delta_k^2 \right].$$

**Proof.** The definitions of  $\tau_k$  and  $z_k$  imply that

$$\langle S_k z_k, [H_k + M_k] S_k z_k \rangle = \langle z_k, [H_k^S + M_k^S] z_k \rangle = \tau_k,$$

from which we deduce that

$$\langle e_i, [H_k^S + M_k^S] z_k \rangle = \tau_k [z_k]_i$$

and therefore, using the identity  $M_k^S = \bar{G}_k$ , that

$$\langle e_i, H_k^S z_k \rangle = \left( \tau_k - \max \left[ [g_k]_i, 0 \right] \right) [z_k]_i$$

for all  $i = 1, \dots, n$ . Hence

$$[z_k]_i = \frac{\langle e_i, H_k^S z_k \rangle}{\delta_i}, \quad \text{where } \delta_i \stackrel{\text{def}}{=} \tau_k - \max \left[ [g_k]_i, 0 \right] \leq \tau_k$$

for all  $i = 1, \dots, n$  such that  $[g_k]_i > 0$ . Now the maximum step along the direction  $S_k z_k$  that preserves sufficient feasibility in the sense of the second part of (13.12.23) is given by

$$t_k^F = \min_{i|[g_k]_i > 0} \frac{(1 - \varsigma_k) [x_k]_i}{[S_k z_k]_i} = \min_{i|[g_k]_i > 0} \frac{(1 - \varsigma_k) [x_k]_i^{\frac{1}{2}}}{[z_k]_i} = \min_{i|[g_k]_i > 0} \frac{(1 - \varsigma_k) \delta_i [x_k]_i^{\frac{1}{2}}}{\langle e_i, H_k^S z_k \rangle}.$$

Let  $j$  be the index in  $\{1, \dots, n\}$  for which this minimum is achieved. Then

$$\left[ t_k^F \right]^2 = \left[ \frac{(1 - \varsigma_k) \delta_j [x_k]_j^{\frac{1}{2}}}{\langle e_j, H_k^S z_k \rangle} \right]^2 = \frac{(1 - \varsigma_k)^2 \delta_j^2 [x_k]_j}{\langle e_j, H_k^S z_k \rangle^2}.$$

But

$$\langle e_j, H_k^S z_k \rangle^2 = z_k^T S_k H_k (S_k e_j e_j^T S_k) H_k S_k z_k = \langle H_k S_k z_k, D_k H_k S_k z_k \rangle,$$

where  $D_k$  is an  $n \times n$  diagonal matrix with

$$[D_k]_{ii} = \begin{cases} [x_k]_j & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, using the Cauchy–Schwarz inequality and the definitions of  $\delta_j$  and  $\varsigma_{\max}$ ,

$$\left[ t_k^F \right]^2 \geq \frac{(1 - \varsigma_{\max})^2 \delta_j^2 [x_k]_j}{\|D_k\| \|H_k S_k z_k\|^2} \geq \frac{(1 - \varsigma_{\max})^2 \tau_k^2}{\langle z_k, [H_k^S]^2 z_k \rangle} = \frac{(1 - \varsigma_{\max})^2}{\kappa_{\text{umh}}^2} \tau_k^2.$$

We now observe that the assumptions of the theorem and the fact that  $S_k z_k$  is a descent direction imply that

$$\langle \nabla_x m_k(x_k + tS_k z_k), S_k z_k \rangle = \langle g_k, S_k z_k \rangle + t \langle z_k, H_k^S z_k \rangle < 0$$

for all  $t \in (0, 1]$ . Hence the minimum in the definition of  $x_k^{\text{CSE}}$  must lie on the boundary of the trust region and/or at a distance corresponding to sufficient feasibility. As a consequence, we obtain that

$$t^{\text{CSE}} = \min \left[ \frac{\Delta_k}{\|S_k z_k\|_k}, t_k^F \right] = \min \left[ \Delta_k, t_k^F \right],$$

since  $\|S_k z_k\|_k = \|S_k^{-1} S_k z_k\| = \|z_k\| = 1$ . Replacing this value in the model's expression and using the descent property of  $S_k z_k$  then gives that

$$\begin{aligned} m_k(x_k) - m_k(x_k^{\text{CSE}}) &= -t_k^{\text{CSE}} \langle g_k, S_k z_k \rangle - \frac{1}{2} [t_k^{\text{CSE}}]^2 \langle z_k, H_k^S z_k \rangle \\ &\geq -\frac{1}{2} [t_k^{\text{CSE}}]^2 \langle z_k, H_k^S z_k \rangle \\ &\geq -\frac{1}{2} |\tau_k| \min \left[ \Delta_k^2, \frac{(1-\varsigma_{\max})^2}{\kappa_{\text{umh}}^2} \tau_k^2 \right], \end{aligned}$$

which then implies the desired bound with  $\kappa_{\text{sod}} \stackrel{\text{def}}{=} \frac{1}{2} \min[1, \frac{(1-\varsigma_{\max})^2}{\kappa_{\text{umh}}^2}]$ .  $\square$

As a consequence of this theorem, we see that assuming AA.2b (with  $\kappa_{\text{lsd}} = \infty$ ) is also reasonable if we require the model decrease to be at least a fraction of that obtained at the constrained scaled eigenpoint.

We now return to our other concern, namely, the difficulty caused by AM.5b, which is that the matrix  $M_k$  does not tend to zero when a first-order critical point is approached, and consequently the Hessian of the model and that of the objective function do not asymptotically coincide as  $g_k^S$  converges to zero. As a result, the ratio  $\rho_k$  in (13.12.10) does not converge to 1, and the proofs of convergence to second-order critical points fail. To circumvent this difficulty, Coleman and Li propose using the technique described in Section 10.4.3, where the model is assumed to consist of a first part  $m_k^f$  augmented by the quadratic positive definite term  $\langle s_k, M_k s_k \rangle$ . The definition of  $\rho_k$ , that is, (13.12.10), is then altered to ensure that it converges to 1 if  $g_k^S$  converges to zero and  $\nabla_{xx} m_k^f$  converges to  $\nabla_{xx} f$  according to (10.4.15) (p. 392). Step 3 of Algorithm 13.12.1 is thus replaced by the following.

**Algorithm 13.12.2: Step 3 of Coleman and Li's algorithm (version 2)**

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$  and define the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) - \frac{1}{2} \langle s_k, M_k s_k \rangle}{m_k(x_k) - m_k(x_k + s_k)}. \quad (13.12.24)$$

Then if  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k$ ; otherwise define  $x_{k+1} = x_k$ .

Applying now the results of Section 10.4.3 in the context of convergence in dual norms, we obtain the following result.

**Theorem 13.12.6** Suppose that AF.1–AF.3 hold for the objective function. Suppose also that its model  $m_k^f$  satisfies AM.1–AM.4 and that

$$\lim_{k \rightarrow \infty} \|S_k(\nabla_{xx} f(x_k) - H_k)S_k\| = 0 \text{ whenever } \lim_{k \rightarrow \infty} \|g_k^s\| = 0. \quad (13.12.25)$$

Suppose finally that the iterates generated by the Coleman–Li algorithm (version 2) remain in a bounded domain of the positive orthant. Then every limit point  $x_*$  of the sequence of iterates is second-order critical.

**Proof.** As mentioned above, AM.6b holds because  $m_k$  is quadratic, and AA.2b is guaranteed by Step 2 of the algorithm (as justified by Theorem 13.12.5). Finally, AM.5b follows for  $m_k^f$  from the definition of the model (13.12.7) and (13.12.25). Hence we may apply the conclusions of Sections 8.1.4 and 10.4.3 and deduce that

$$\lim_{k \rightarrow \infty} \lambda_{\min} \left[ S_k(\nabla_{xx} f(x_k) + M_k)S_k \right] = \lim_{k \rightarrow \infty} \lambda_{\min} \left[ [\nabla_{xx} f(x_k)]^s + M_k^s \right] \geq 0,$$

which in turn implies the desired conclusion because of Theorem 13.12.2.  $\square$

This concludes the convergence analysis for the method of Coleman and Li. We note that the modification of Step 3 has not clarified the question of which function  $m_k(x_k + s)$  is a model, since the numerator of the right-hand side of (13.12.24) may no longer be interpreted as the reduction achieved on a given function. However, despite this blurred interpretation, the method remains theoretically attractive because it does not require an outer iteration.

## Notes and References for Section 13.12

The algorithm discussed in this section is that of Coleman and Li (1996a, 1998a) recast in the simpler context of pure nonnegativity constraints. The authors' original proposition is for general bound constraints. In this case, the scaling function  $v(x)$  is defined componentwise by

$$[v(x)]_i = \begin{cases} [x]_i - [x_\ell]_i & \text{if } [\nabla_x f(x)]_i > 0 \text{ and } [x_\ell]_i > -\infty, \\ [x]_i - [x_u]_i & \text{if } [\nabla_x f(x)]_i < 0 \text{ and } [x_u]_i < \infty, \\ +1 & \text{if } [\nabla_x f(x)]_i > 0 \text{ and } [x_\ell]_i = -\infty, \\ -1 & \text{if } [\nabla_x f(x)]_i < 0 \text{ and } [x_u]_i = \infty, \end{cases}$$

with the convention that if  $[\nabla_x f(x)]_i = 0$  for some  $i$ , then  $[J(x)]_{ii} = 0$ . We then obtain that  $\bar{G}_k = |G_k|$ . A code using this approach is included in Coleman, Branch, and Grace's (1999) MATLAB *Optimization Toolbox*.

The reformulation of the optimality conditions as a system of nonlinear equations (13.12.3) was earlier proposed by the same authors in the context of a “reflexive” linesearch algorithm for the bound-constrained problem. This reformulation appeared in Coleman and Li (1994).

See also Li (1993), Branch, Coleman, and Li (1999), Coleman and Li (1996b), and Dennis and Vicente (1996) for some additional motivation for these techniques. The paper of Ulbrich and Ulbrich (1997) analyses Coleman and Li's algorithm in the context of infinite-dimensional problems with pointwise bounds. More general scaling techniques, which still fit in the same framework, are proposed. This idea is developed further by Heinkenschloss, Ulbrich, and Ulbrich (1999), who additionally propose enforcing feasibility by projecting the step onto the feasible domain instead of using the simple scaling proposed by Coleman and Li. Fast local convergence is then obtained even without assuming strict complementarity at the limit point. Extension of the method to nonquadratic models is clearly possible, since this level of generality is supported by Sections 8.1.2 and 8.1.3. From the theoretical point of view, it is indeed only necessary to modify (13.12.25) to ensure that the convergence of the Hessians occurs uniformly in the trust region, as in AM.5b. Finally, we note that a variant of Coleman and Li's algorithm is discussed in Yin and Han (1998), in which the trust region is intersected with the feasible domain.

Coleman and Li (1997) extend their algorithm for the case where the constraints are linear inequalities, that is, of the form

$$Ax \geq b,$$

where  $A$  is an  $m \times n$  matrix and  $b \in \mathbb{R}^m$ . We briefly review this method here. As was the case above, we assume that  $\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$  has a nonempty relative interior. The key to the development of Section 13.12 was to reformulate the first-order necessary condition of the constrained problem as a nonlinear system of equations (see Theorem 13.12.1), and the same approach is also used in this more general context. Specifically, these conditions may be written as

$$\nabla_x f(x) - A^T y = 0, \quad \text{diag}(Ax - b)y = 0, \quad Ax \geq b \text{ and } y \geq 0. \quad (13.12.26)$$

As the algorithm ensures that the third of these conditions (primal feasibility) will be satisfied for all iterates, we now focus on the first two, leaving the discussion of the fourth (which along with the first constitutes dual feasibility) for later. If we define

$$C(x) = \text{diag}(Ax - b),$$

then these first two conditions become

$$\nabla_x f(x) - A^T y = 0 \text{ and } C(x)y = 0.$$

Writing an iteration of Newton's method for this system then gives the system

$$\begin{pmatrix} \nabla_{xx} f(x_k) & A^T \\ Y_k A & -C_k \end{pmatrix} \begin{pmatrix} s_k \\ y_k - \bar{y}_{k+1} \end{pmatrix} = -\begin{pmatrix} \nabla_x f(x_k) - A^T y_k \\ C_k y_k \end{pmatrix}, \quad (13.12.27)$$

where we have set  $Y_k \stackrel{\text{def}}{=} \text{diag}([y_k]_1, \dots, [y_k]_m)$  and  $C_k \stackrel{\text{def}}{=} C(x_k)$ . This defines a step often called the *affine step* in primal-dual interior-point methods. If we now define  $v_k = C_k^{-\frac{1}{2}} As_k \in \mathbb{R}^m$ , that is,

$$As_k - C_k^{\frac{1}{2}} v_k = 0, \quad (13.12.28)$$

the second equation of (13.12.27) becomes, after premultiplication by  $C_k^{-\frac{1}{2}}$ ,

$$C_k^{-\frac{1}{2}} Y_k C_k^{\frac{1}{2}} v_k - C_k^{\frac{1}{2}} (y_k - \bar{y}_{k+1}) = Y_k v_k - C_k^{\frac{1}{2}} (y_k - \bar{y}_{k+1}) = -C_k^{\frac{1}{2}} y_k \quad (13.12.29)$$

because  $Y_k$  is diagonal. Substituting (13.12.28) and (13.12.29) into (13.12.27) then gives that

$$\begin{pmatrix} \nabla_{xx}f(x_k) & 0 & A^T \\ 0 & Y_k & -C_k^{\frac{1}{2}} \\ A & -C_k^{\frac{1}{2}} & 0 \end{pmatrix} \begin{pmatrix} s_k \\ v_k \\ y_k - \bar{y}_{k+1} \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x_k) - A^T y_k \\ C_k^{\frac{1}{2}} y_k \\ 0 \end{pmatrix}. \quad (13.12.30)$$

But this system also corresponds to the Newton step for the problem

$$\min_{s,v} \langle \nabla_x f(x_k), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} f(x_k) s \rangle + \frac{1}{2} \langle v, Y_k v \rangle \quad (13.12.31)$$

subject to

$$As - C_k^{\frac{1}{2}} v = 0, \quad (13.12.32)$$

whenever both  $\nabla_{xx} f(x_k)$  and  $Y_k$  are positive semidefinite. As this problem involves the Taylor series of  $f(x_k + s)$ , we should expect that it is only meaningful in some trust region around the current iterate. We therefore consider the problem (13.12.31)–(13.12.32) augmented with the constraint that

$$\left\| \begin{pmatrix} s \\ v \end{pmatrix} \right\| \leq \Delta_k \quad (13.12.33)$$

for some  $\Delta_k > 0$ . Finally, we observe that the variable  $v$  can be eliminated from (13.12.32), and problem (13.12.31)–(13.12.33) then becomes

$$\min_s m_k(x_k + s) \stackrel{\text{def}}{=} m_k^f(x_k + s) + \frac{1}{2} \langle As, [C_k^{-1} Y_k] As \rangle \stackrel{\text{def}}{=} m_k^f(x_k + s) + \frac{1}{2} \langle s, M_k s \rangle \quad (13.12.34)$$

subject to

$$\|s\|_k \stackrel{\text{def}}{=} \|s\|_{I + A^T C_k^{-1} A} \leq \Delta_k n, \quad (13.12.35)$$

where we have replaced the objective function  $f(x_k + s)$  by its quadratic model  $m_k^f(x_k + s)$ , which is supposed to satisfy AM.1–AM.3 and AM.4i. On the other hand, if some multiplier  $[y_k]_i$  is negative, Coleman and Li propose considering its absolute value, so that the matrix  $Y_k$  is replaced by  $|Y_k|$  in all formulae above. In particular, the expression for the matrix  $M_k$  becomes

$$M_k = A^T C_k^{-1} |Y_k| A.$$

This modification is in line with the idea of primal-dual methods, where nonnegativity of the dual variables is maintained throughout the calculation. However, the replacement of  $Y_k$  by  $|Y_k|$  in (13.12.30) makes the interpretation of this system as Newton's method for the first-order conditions slightly tenuous.

One then obtains an algorithm where a step  $s_k$  approximately solves the problem given by (13.12.34)–(13.12.35) for a suitable choice of the Lagrange multipliers  $y_k$ . This algorithm is called TRAM (trust-region affine-scaling method) by its authors. It is similar in structure to Coleman and Li's algorithm (with Step 3 as on p. 565). Sufficient model decrease is then established by defining a suitably modified Cauchy point along the direction

$$p_k = - \left( \nabla_x f(x_k) - A^T y_k \right), \quad \text{where } y_k = \arg \min_y \left\| \begin{pmatrix} A^T \\ -C_k^{\frac{1}{2}} \end{pmatrix} y - \begin{pmatrix} \nabla_x f(x_k) \\ 0 \end{pmatrix} \right\|. \quad (13.12.36)$$

Note that  $p_k$  is restricted to the null-space of the matrix  $(A - C_k^{\frac{1}{2}})$ . Coleman and Li then prove that, provided the constraints active at any  $x \in \mathcal{C}$  are linearly independent and the iterates stay bounded, convergence must occur to a point satisfying the first three conditions of (13.12.26). However, in order to obtain that the limit points are first-order critical, one still

has to prove that  $y(x_*) \geq 0$ , in which case the desired identity  $Y_k = |Y_k|$  is asymptotically obtained. In order to obtain dual feasibility, Coleman and Li propose a further modification of the algorithm. Since the iterates could, without this modification, converge to a noncritical point satisfying the first three parts of (13.12.26), they suggest altering the definition of the trust-region scaling when such a point is approached. The modified algorithm can then be proved to be globally convergent to first-order critical points. Furthermore, second-order convergence as well as rate of convergence results may also be obtained if the Hessian of the objective function and of its model  $m_k^f$  asymptotically coincide and if the trial step produces at least a fraction of the best possible decrease of the model in the intersection of the trust region and the feasible domain of (13.12.9).

Coleman and Li (1998b) also suggest a way to remove the proposed modification of the scaling matrix in the case where noncritical points satisfying the first three parts of (13.12.26) are approached: the idea is to compute the dual variables  $y_k$  using a constrained least-squares procedure, that is, by imposing nonnegativity constraints in (13.12.36). Very recently, Coleman and Li (2000) have extended their algorithm to handle general nonlinear constraints.

## **Part IV**

---

# **Trust-Region Methods for General Constrained Optimization and Systems of Nonlinear Equations**

In this, the final main part of the book, we turn to methods that are appropriate when the constraints are both nonlinear and nonconvex.

---

# Chapter 14

---

## Penalty-Function Methods

---

### 14.1 Penalty Functions and Constrained Optimization

We now turn our attention to the general nonlinear programming problem. One way that comes readily to mind if we wish to handle constrained problems via trust-region methods is to incorporate the more complicated constraints into the objective function. Thus the trust-region method has to handle only the more tractable constraints directly. The two usual ways to do this involve penalty and barrier methods. We have already described the latter in Chapter 13.

Thus we will consider how to find an  $x$  to

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \text{ for } i \in \mathcal{E} \\ & && \text{and} \quad c_i(x) \geq 0 \text{ for } i \in \mathcal{I}, \end{aligned} \tag{14.1.1}$$

by solving a sequence of unconstrained problems. Here  $\mathcal{E}$  and  $\mathcal{I}$  are disjoint sets of the indices of equality and inequality constraints, and  $f$  and the  $c_i$  map  $\mathbb{R}^n$  into  $\mathbb{R}$ . The reader should recall that we considered optimality conditions for this problem in Section 3.2.2. In the preceding chapters, we have considered special cases where the constraints were either linear or convex (in Section 13.9, we noted that barrier algorithms easily generalize to the case where any inequalities are nonlinear). We now allow any or all of the constraints to be both nonconvex and nonlinear.

For simplicity, throughout this chapter we shall, by and large, concentrate on the problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \text{ for } i \in \mathcal{E}, \end{aligned} \tag{14.1.2}$$

which only involves equality constraints. We shall, when it is appropriate, indicate extensions to the case where inequalities are also present.

The general constrained optimization problem (14.1.1) introduces a number of difficulties, at least one of which we have not seen before. Namely, rather than having

a single objective (“minimize the objective function”), we now have two, often conflicting requirements, those of minimizing  $f(x)$  and of simultaneously satisfying the constraints. Certainly this is not an issue for unconstrained minimization, while for many methods with special classes of constraints, such as linear or convex constraints, it is often possible to ensure that once a feasible point has been found, all subsequent iterates stay feasible, and so the algorithm is free to concentrate on the minimization. Not so for nonconvex, nonlinear constraints, where it is often both difficult—or at least costly—and undesirable to ensure that a constraint that was feasible at one iterate will also be feasible at the next. In principle, all that is required is that iterates converge to a point that is both feasible and optimal, and it doesn’t really matter what happened along the way.<sup>231</sup> Thus we see that we are now faced with *two* requirements, while we have been used to only dealing with a single objective.

Penalty-function methods are a way to resolve this dichotomy of objectives. The basic idea is to try to encompass in a *single* objective the requirements of both optimality and feasibility. Optimality is normally represented by the objective function  $f(x)$ , while feasibility is measured by some norm of a componentwise function of the distance to feasibility<sup>232</sup>  $\phi(c(x), p)$ . The function  $\phi$  may depend upon a number of auxiliary parameters  $p$  and is normally required to vanish at the origin (i.e.,  $\phi(0, p) = 0$ ) and to be infinite at infinity (i.e.,  $\lim_{\alpha \rightarrow \pm\infty} \phi(\alpha, p) = +\infty$ ). Different combinations of norms,  $\phi$ ’s, and  $p$ ’s, produce feasibility measures with different properties.

The feasibility and optimality measures are combined in a single *penalty* function, which is a weighted combination of the two. Thus, for the problem (14.1.2), the generic penalty function may be expressed as

$$\Phi(x, w, p) = f(x) + \|W\phi(c(x), p)\|, \quad (14.1.3)$$

where  $w$  is a vector of weights and  $W$  is the diagonal matrix formed from  $w$ . The idea now is simply to approximately minimize  $\Phi$  for a sequence of values of the weights  $w$  and parameters  $p$ . These are *unconstrained* minimizations, and the methods we considered in Part 2 of the book are, in principle, appropriate. However, we should caution the reader at this stage that some surprises lie in store.

Intuitively, if the weights are allowed to grow, an increasing part of the minimization of  $\Phi$  will be directed towards keeping the contribution  $\phi(c(x), p)$  in (14.1.3) small. This in turn will force the constraints ever closer to feasibility. Indeed, if the weights were infinite, the minimization of  $\Phi$  would effectively ensure feasibility. For finite, but large, weights, the contribution of  $f$  to  $\Phi$  means that some effort will also be directed to reducing  $f$ . Thus one might expect that the limits of minimizers of  $\Phi$  as the weights approach infinity would give suitable solutions to (14.1.2), and that is normally the case.

---

<sup>231</sup>Practitioners do not necessarily agree with this philosophy, arguing that it is better to be at a suboptimal but feasible point when their computing resources are exhausted, than at an infeasible point with a low function value. Moreover, it is not unusual to have functions that cannot be evaluated at some infeasible points.

<sup>232</sup>When the constraints are inequalities, the distance to feasibility is  $\phi(\min[c(x), 0], p)$ .

This is but one example of how penalty-function methods work and is best exemplified by the quadratic penalty function we shall consider in Section 14.3. However, there are other, more subtle approaches. One, as typified by the augmented Lagrangian function we consider in Section 14.4, achieves convergence by juggling the parameters in  $\phi$  without resorting to infinite weights. A second, which forms the basis for Sections 14.5 and 14.6, merely requires that the weights be larger than some—unfortunately *a posteriori*—values without the need for additional parameters.

Before we start, we define that common notation that will occur throughout this chapter. Although some of the results may be established under weaker conditions, it is convenient to make the general assumption that

**AW.1** the functions  $f(x)$  and  $c_i(x)$  are twice-continuously differentiable functions of  $x$ ,

and thus both AF.1 and AC.1 hold throughout. We also let  $c(x)$  be the vector whose components are the  $c_i(x)$  for  $i \in \mathcal{E}$  and denote the Jacobian of  $c(x)$  by  $A(x)$ . Furthermore, we shall denote the cardinality of  $\mathcal{E}$  by  $m$ . Finally, we define the Lagrangian function

$$\ell(x, y) = f(x) - \sum_{i \in \mathcal{E}} y_i c_i(x), \quad (14.1.4)$$

where  $y$  is any set of auxiliary variables, and write the augmented matrix

$$K(x, y, \mu) = \begin{pmatrix} \nabla_{xx} \ell(x, y) & A^T(x) \\ A(x) & -\mu I \end{pmatrix}.$$

As usual, a subscript  $k$  attached to any of these quantities will be used to denote that quantity evaluated with appropriate arguments  $x_k$  and  $y_k$ .

## 14.2 Smooth Penalty Functions

The first penalty function to be widely used was the *quadratic* penalty function. Given a strictly positive *penalty parameter*  $\mu$ , the quadratic penalty function  $\Phi_0$  is defined to be<sup>233</sup>

$$\Phi_0(x, \mu) = f(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i(x)^2 = f(x) + \frac{1}{2\mu} \|c(x)\|_2^2. \quad (14.2.1)$$

More generally, we can allow more flexibility by considering a related penalty function called the *augmented Lagrangian* function

$$\begin{aligned} \Phi_y(x, \mu) &= f(x) - \sum_{i \in \mathcal{E}} c_i(x)y_i + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i(x)^2 \\ &= f(x) - \langle c(x), y \rangle + \frac{1}{2\mu} \|c(x)\|_2^2, \end{aligned} \quad (14.2.2)$$

where  $y_i$ ,  $i \in \mathcal{E}$ , are given constants. As its name implies, (14.2.2) may be viewed as an augmentation or shift of the Lagrangian function (14.1.4), in which the augmenting

---

<sup>233</sup>The constant 2 is merely included so that derivatives of  $\Phi_0$  have a convenient form.

terms add a quadratic penalty of the constraint violation. Alternatively, (14.2.2) might equally well be regarded as an augmentation of the quadratic penalty function (14.2.1). In any event, there is nothing to be lost in this section by considering the augmented Lagrangian rather than the quadratic penalty function, particularly as the latter may be recovered from the former with the special choice  $y = 0$ . It is important to note that the functions (14.2.1) and (14.2.2) are at least as smooth<sup>234</sup> as  $f$  and the  $c_i$ . There are other smooth penalty functions, such as the quartic penalty function

$$f(x) + \frac{1}{4\mu} \sum_{i \in \mathcal{E}} c_i(x)^4$$

or, in the case of inequality constraints, the exponential penalty function

$$f(x) + \mu \sum_{i \in \mathcal{I}} \left( e^{-c_i(x)/\mu} - 1 \right)$$

for which essentially identical results may be established, but as there appear to be no compelling reasons to prefer them, we shall consider them no further.

We recall the optimality conditions from Section 3.2.2 that a point  $x_*$  is a first-order critical point for the problem (14.1.2) if there is a vector of Lagrange multipliers  $y_*$  for which

$$\nabla_x f(x(\mu)) - A^T(x_*)y_* = 0 \quad \text{and} \quad c(x_*) = 0. \quad (14.2.3)$$

We now wish to draw an analogy between these conditions and the suggestion we made in the previous section that we should aim to minimize  $\Phi_y(x, \mu)$  for a sequence of fixed  $y$  and  $\mu$ . If  $x(\mu)$  is such a minimizer and  $\Phi \in C^1$ , on differentiating  $\Phi$  and applying Theorem 3.2.1 (p. 38), we automatically have that

$$\nabla_x \Phi_y(x(\mu), \mu) = \nabla_x f(x(\mu)) - A^T(x(\mu))y_y^F(x(\mu), \mu) = 0, \quad (14.2.4)$$

where

$$y_y^F(x, \mu) = y - \frac{c(x)}{\mu}. \quad (14.2.5)$$

Comparing (14.2.3) and (14.2.4), it is apparent that  $y_y^F(x(\mu), \mu)$  might be a useful approximation to  $y_*$ . In addition, multiplying both sides of (14.2.5) by  $\mu$ , we see that

$$c(x(\mu)) = \mu(y - y_y^F(x(\mu), \mu)). \quad (14.2.6)$$

The relationship (14.2.6) is highly suggestive, as it indicates that  $c(x(\mu))$  may be forced to zero by either driving  $\mu$  to zero or resetting  $y$  to  $y_y^F(x(\mu), \mu)$  (or both). We shall consider the first of these possibilities in Section 14.3, deferring the other possibility to Section 14.4.

---

<sup>234</sup>They may actually be smoother. For instance, the quadratic penalty function is differentiable for constraint functions with discontinuous derivatives along the surface  $c(x) = 0$ .

The function  $y_y^F(x, \mu)$  is known as a *first-order Lagrange multiplier estimate*, and these estimates always satisfy the important identity

$$\nabla_x \Phi_y(x, \mu) = \nabla_x \ell(x, y_y^F(x, \mu)). \quad (14.2.7)$$

There are other useful Lagrange multiplier estimates. Of particular interest, at least from a theoretical point of view, are the *least-squares Lagrange multiplier estimates*  $y^{LS}(x)$ , which are defined to be the minimum-norm solution of

$$\min_y \|\nabla_x f(x) - A^T(x)y\|_2$$

(see Sections 2.3 and 12.4.1). The solution is thus

$$y^{LS}(x) = (A^+(x))^T g(x) \quad (14.2.8)$$

at all points where the right generalized inverse

$$A^+(x) = A^T(x)(A(x)A^T(x))^{-1}$$

of  $A(x)$  is well defined. Clearly, at  $x_*$ ,  $y^{LS}(x_*) = y_*$ , and thus it is plausible that  $y^{LS}(x)$  will prove to be a useful approximation to  $y_*$  when  $x$  is close to  $x_*$ . We note that  $y^{LS}(x)$  is almost always differentiable, and its derivative is given in the following lemma.

**Lemma 14.2.1** Suppose that AW.1 holds. If  $A(x)A^T(x)$  is nonsingular,  $y^{LS}(x)$  is differentiable and its derivative is given by

$$\nabla_x y^{LS}(x) = (A^+(x))^T \nabla_{xx} \ell(x, y^{LS}(x)) + (A(x)A^T(x))^{-1} E(x), \quad (14.2.9)$$

where the  $i$ th row of  $E(x)$  is  $(g(x) - A(x)^T y^{LS}(x))^T \nabla_{xx} c_i(x)$ .

**Proof.** The result follows by observing that (14.2.8) may be rewritten as

$$r(x) + A^T(x)y^{LS}(x) = g(x) \quad \text{and} \quad A(x)r(x) = 0 \quad (14.2.10)$$

for some vector  $r(x)$ . Differentiating (14.2.10) and eliminating the derivative of  $r(x)$  from the resulting equations gives the required result.  $\square$

We now try to formalize the arguments sketched in the opening paragraphs of this section. To do so, we shall consider a sequence of strictly positive penalty parameters  $\{\mu_k\}$ , any sequence of vectors  $\{y_k\}$ , and any sequence of iterates  $\{x_k\}$  that are generated so as to be approximate minimizers of  $\Phi_{y_k}$  in the sense that

$$\|\nabla_x \Phi_{y_k}(x_k, \mu_k)\| \leq \omega_k \quad (14.2.11)$$

for some sequence of positive scalars  $\{\omega_k\}$ ; for the purposes of Section 14.3, it suffices that  $y_k = y$  for all  $k$ , but more general sequences will be encountered in Section 14.4.

We also assume that the iterates lie in a compact region, that is, that AI.1 holds. This condition is really needed to ensure that the functions stay bounded, but of course it has the additional consequence that the infinite sequence  $\{x_k\}$  will have at least one limit point. Such an assumption is quite strong in practice, but in its absence it is entirely possible that the function  $\Phi_y$  is unbounded from below for any value of the penalty parameter.

We also require a second, strong assumption that is quite characteristic of simple penalty functions. As before, we let  $\mathcal{L}_*$  be the set of all limit points of the sequence  $\{x_k\}$  and require that AO.1c holds, that is, that the gradients of the constraints at such limit points are linearly independent. This assumption appears to be important both in theory and in practice for simple penalty functions. As we mentioned in Section 3.2.2, this assumption is actually a first-order constraint qualification. Without AO.1c, it is easy to find examples whose limit point is not a first-order critical point for (14.1.2). Indeed, suppose that  $f(x) \equiv 0$ , that  $A(x_*)$  is rank-deficient, and that  $c(x_*) \neq 0$  lies in the null-space of  $A^T(x_*)$ . Then  $\nabla_x \Phi_0(x_*, \mu) = 0$  for all  $\mu$ , but  $x_*$  is not feasible. Thus AO.1c is essential for penalty methods based upon  $\Phi_y$ , and this somewhat limits the power of such methods.

We can also improve upon the result we are about to establish if we make stronger assumptions on the problem functions.

**AW.1b** The second derivatives of the functions  $f(x)$  and  $c_i(x)$  exist and are Lipschitz continuous for all points in  $\Omega$ .

**AO.3b** The augmented matrix  $K(x_*, y^{\text{LS}}(x_*), 0)$  is nonsingular for all  $x_* \in \mathcal{L}_*$ .

We note that AO.1c is implied by AO.3b. As a precursor to the main results in the next sections, we have the following general convergence result.

**Lemma 14.2.2** Suppose that AW.1 holds. Let  $\{x_k\}, k \in \mathcal{K}$ , be a sequence satisfying AI.1, which converges to the point  $x_*$  for which AO.1c holds, and let  $y_* = y^{\text{LS}}(x_*)$ , where  $y^{\text{LS}}$  is given by (14.2.8). Assume that  $\{y_k\}, k \in \mathcal{K}$ , is any sequence of vectors and that  $\{\omega_k\}, k \in \mathcal{K}$ , form a bounded sequence of positive scalars. Suppose further that (14.2.11) holds, where the  $\omega_k$  are positive scalar parameters that converge to zero as  $k \in \mathcal{K}$  increases. Then

(i) there are positive constants  $\kappa_1, \kappa_2$ , and an integer  $k_0$  such that

$$\|y_k^F - y_*\| \leq \kappa_1 \omega_k + \kappa_2 \|x_k - x_*\|, \quad (14.2.12)$$

$$\|y_k^{\text{LS}} - y_*\| \leq \kappa_2 \|x_k - x_*\|, \quad (14.2.13)$$

and

$$\|c(x_k)\| \leq \kappa_1 \omega_k \mu_k + \mu_k \|y_k - y_*\| + \kappa_2 \mu_k \|x_k - x_*\|, \quad (14.2.14)$$

where  $y_k^F = y_{y_k}^F(x_k, \mu_k)$  and  $y_k^{LS} = y^{LS}(x_k)$  for all  $k \geq k_0$  ( $k \in \mathcal{K}$ ).

Suppose, in addition, that

$$c(x_*) = 0. \quad (14.2.15)$$

Then

- (ii)  $x_*$  is a first-order critical point for the problem (14.1.1),  $y_*$  is the corresponding vector of Lagrange multipliers, and the sequences  $\{y_k^F\}$  and  $\{y_k^{LS}\}$  converge to  $y_*$  for  $k \in \mathcal{K}$ .

Suppose in addition to (14.2.15) that we replace AO.1c and AW.1, respectively, by AO.3b and AW.1b. Then

- (iii) there are constants  $\mu_{\max}$ ,  $\kappa_3 - \kappa_7$ , and  $k_1$  so that, if  $\mu_k \leq \mu_{\max}$ ,

$$\|x_k - x_*\| \leq \kappa_3 \omega_k + \kappa_4 \mu_k \|y_k - y_*\|, \quad (14.2.16)$$

$$\|y_k^F - y_*\| \leq \kappa_5 \omega_k + \kappa_6 \mu_k \|y_k - y_*\|, \quad (14.2.17)$$

$$\|y_k^{LS} - y_*\| \leq \kappa_7 \omega_k + \kappa_8 \mu_k \|y_k - y_*\|,$$

and

$$\|c(x_k)\| \leq \kappa_5 \mu_k \omega_k + (\mu_k + \kappa_6 \mu_k^2) \|y_k - y_*\| \quad (14.2.18)$$

for all  $k \geq k_1 \in \mathcal{K}$ .

Finally, suppose additionally that

$$\|c(x_k)\| \leq \eta_k$$

for another sequence  $\{\eta_k\}$  which converges to zero as  $k \in \mathcal{K}$  increases. Then

- (iv) the sequences  $\{x_k\}$ ,  $\{y_k^F\}$ , and  $\{y_k^{LS}\}$  converge to their limits at an R-rate at least as fast as the slower of the Q-rates of  $\{\omega_k\}$  and  $\{\eta_k\}$ .

**Proof.** As a consequence of AO.1c, AW.1, and AI.1, we have that for  $k \in \mathcal{K}$  sufficiently large,  $A^+(x_k)$  exists, is bounded, and converges to  $A^+(x_*)$ . Thus we may write

$$\|(A^+(x_k))^T\| \leq \kappa_1 \quad (14.2.19)$$

for some constant  $\kappa_1 > 0$ . Then (14.2.7) and the inner iteration termination criterion (14.2.11) give that

$$\|g(x_k) - A^T(x_k)y_k^F\| \leq \omega_k. \quad (14.2.20)$$

By assumption,  $y^{LS}(x)$  is bounded for all  $x$  in a neighbourhood of  $x_*$ . Thus we may

deduce from (14.2.8), (14.2.19), and (14.2.20) that

$$\begin{aligned}\|y_k^F - y^{\text{LS}}(x_k)\| &= \|(A^+(x_k))^T g(x_k) - y_k^F\| \\ &= \|(A^+(x_k))^T(g(x_k) - A^T(x_k)y_k^F)\| \\ &\leq \|(A^+(x_k))^T\| \omega_k \leq \kappa_1 \omega_k.\end{aligned}\quad (14.2.21)$$

Moreover, from the integral mean-value theorem (Theorem 3.1.3 [p. 28]) and Lemma 14.2.1 we have that

$$y^{\text{LS}}(x_k) - y^{\text{LS}}(x_*) = \int_0^1 \nabla_x y^{\text{LS}}(x(s)) ds \cdot (x_k - x_*), \quad (14.2.22)$$

where  $\nabla_x y^{\text{LS}}(x)$  is given by equation (14.2.9) and where  $x(s) = x_k + s(x_* - x_k)$ . Now the terms within the integral sign are bounded for all  $x$  sufficiently close to  $x_*$ , and hence (14.2.22) gives

$$\|y^{\text{LS}}(x_k) - y_*\| \leq \kappa_2 \|x_k - x_*\| \quad (14.2.23)$$

for some constant  $\kappa_2 > 0$ , which is just the inequality (14.2.13). We then have that  $y^{\text{LS}}(x_k)$  converges to  $y_*$ . Combining (14.2.21) and (14.2.23) we obtain

$$\|y_k^F - y_*\| \leq \|y_k^F - y^{\text{LS}}(x_k)\| + \|y^{\text{LS}}(x_k) - y_*\| \leq \kappa_1 \omega_k + \kappa_2 \|x_k - x_*\|, \quad (14.2.24)$$

the required inequality (14.2.12). Since by assumption  $\omega_k$  tends to zero as  $k$  increases, (14.2.12) implies that  $y_k^F$  converges to  $y_*$ , and from (14.2.20) we have that

$$\nabla_x \ell(x_*, y_*) = g(x_*) - A^T(x_*)y_* = 0. \quad (14.2.25)$$

Furthermore, multiplying (14.2.5) by  $\mu_k$ , we obtain

$$c(x_k) = \mu_k((y_k^F - y_*) + (y_* - y_k)). \quad (14.2.26)$$

Taking norms of (14.2.26) and using (14.2.24) we derive (14.2.14), which concludes the proof of assertion (i).

Now suppose that (14.2.15) also holds. Then (14.2.25) and (14.2.15) imply that  $x_*$  is a first-order critical point and  $y_*$  are a corresponding set of Lagrange multipliers. Moreover, (14.2.12) and (14.2.13) ensure the convergence of the sequences  $\{y_k^F\}$  and  $\{y_k^{\text{LS}}\}$  to  $y_*$  for  $k \in \mathcal{K}$ , and thus assertion (ii) is true.

To prove assertion (iii), we note that (i) and (ii) remain true under AO.3b and AW.1b since these imply AO.1c and AW.1. To obtain the estimate (14.2.16), we observe that AW.1b implies that the Hessian of the Lagrangian function has bounded, Lipschitz continuous second derivatives in  $\Omega$ , since this is true for each individual component. We consider a perturbation of the gradient of the Lagrangian function about  $(x_*, y_*)$ . We then have

$$\begin{aligned}\nabla_x \ell(x_k, y_k^F) &= \nabla_x \ell(x_*, y_*) + \nabla_{xx} \ell(x_*, y_*)(x_k - x_*) \\ &\quad + \nabla_{yx} \ell(x_*, y_*)(y_k^F - y_*) + r_x(x_k - x_*, y_k^F - y_*)\end{aligned}$$

and

$$\begin{aligned}\nabla_y \ell(x_k, y_k^F) &= \nabla_y \ell(x_*, y_*) + \nabla_{xy} \ell(x_*, y_*)(x_k - x_*) \\ &\quad + \nabla_{yy} \ell(x_*, y_*)(y_k^F - y_*) + r_y(x_k - x_*, y_k^F - y_*).\end{aligned}$$

Now Theorem 3.1.6 (p. 29) implies that the composite vector  $r(x, y) = (r_x^T(x, y), r_y^T(x, y))^T$  satisfies the bound

$$\|r(x, y)\| \leq \gamma \|(x, y)\|^2 \quad (14.2.27)$$

for some global Lipschitz constant  $\gamma$ . Evaluating the various components of the derivatives of  $\ell(x, y)$  (from which we find that  $\nabla_{yy} \ell(x, y) = 0$ ,  $\nabla_{xy} \ell(x, y)$  is just  $A^T(x)$ , and  $\nabla_y \ell(x, y)$  is  $c(x)$ ), and recalling that the first-order criticality of  $(x_*, y_*)$  implies that  $\nabla_x \ell(x_*, y_*) = 0$  and  $\nabla_y \ell(x_*, y_*) = 0$ , we deduce that

$$\begin{pmatrix} \nabla_x \ell(x_k, y_k^F) \\ c(x_k) \end{pmatrix} = \begin{pmatrix} \nabla_{xx} \ell(x_*, y_*) & A^T(x_*) \\ A(x_*) & 0 \end{pmatrix} \begin{pmatrix} x_k - x_* \\ y_* - y_k^F \end{pmatrix} + r(x_k - x_*, y_k^F - y_*). \quad (14.2.28)$$

Since AO.3b guarantees that  $K(x_*, y_*, 0)$  is nonsingular, we may multiply (14.2.28) by its inverse and take norms, which reveals that

$$\left\| \begin{pmatrix} x_k - x_* \\ y_* - y_k^F \end{pmatrix} \right\| \leq \kappa_{\text{bik}} \left( \left\| \begin{pmatrix} \nabla_x \ell(x_k, y_k^F) \\ c(x_k) \end{pmatrix} \right\| + \gamma \left\| \begin{pmatrix} x_k - x_* \\ y_* - y_k^F \end{pmatrix} \right\|^2 \right), \quad (14.2.29)$$

where  $\kappa_{\text{bik}}$  is a bound on the inverse of  $K(x_*, y_*, 0)$  and we have used the bound (14.2.27). Since  $\{x_k\}$  and  $\{y_k^F\}$ ,  $k \in \mathcal{K}$ , converge to  $x_*$  and  $y_*$ , respectively, we must have that

$$\left\| \begin{pmatrix} x_k - x_* \\ y_k^F - y_* \end{pmatrix} \right\| \leq \frac{1}{2\kappa_{\text{bik}}\gamma}$$

for all  $k \geq k_2 \in \mathcal{K}$  and some  $k_2 \geq 0$  sufficiently large. In this case, (14.2.29) becomes

$$\left\| \begin{pmatrix} x_k - x_* \\ y_k^F - y_* \end{pmatrix} \right\| \leq \kappa_{\text{bik}} \left\| \begin{pmatrix} \nabla_x \ell(x_k, y_k^F) \\ c(x_k) \end{pmatrix} \right\| + \frac{1}{2} \left\| \begin{pmatrix} x_k - x_* \\ y_k^F - y_* \end{pmatrix} \right\|,$$

which may be rearranged to give

$$\left\| \begin{pmatrix} x_k - x_* \\ y_k^F - y_* \end{pmatrix} \right\| \leq 2\kappa_{\text{bik}} \left\| \begin{pmatrix} \nabla_x \ell(x_k, y_k^F) \\ c(x_k) \end{pmatrix} \right\|. \quad (14.2.30)$$

Recalling that (14.2.7) and (14.2.11) imply that

$$\|\nabla_x \ell(x_k, y_k^F)\| \leq \omega_k,$$

and using (14.2.14) and (14.2.30), it then follows that

$$\|x_k - x_*\| \leq 2\kappa_{\text{bik}} (\omega_k + \kappa_1 \omega_k \mu_k + \mu_k \|y_k - y_*\| + \kappa_2 \mu_k \|x_k - x_*\|). \quad (14.2.31)$$

Now suppose that

$$\mu_k \leq \mu_{\max} \stackrel{\text{def}}{=} \min \left[ \frac{1}{\kappa_1}, \frac{1}{4\kappa_{\text{bik}}\kappa_2} \right]. \quad (14.2.32)$$

Then (14.2.31) and (14.2.32) imply that

$$\|x_k - x_*\| \leq 2\kappa_{\text{bik}} (2\omega_k + \mu_k \|y_k - y_*\|) + \frac{1}{2} \|x_k - x_*\|$$

for all  $k \geq k_1 \in \mathcal{K}$ , where  $k_1 \geq \max[k_0, k_2]$ , from which it follows that

$$\|x_k - x_*\| \leq 4\kappa_{\text{bik}} (2\omega_k + \mu_k \|y_k - y_*\|)$$

for all such  $k$ . On defining  $\kappa_3 = 8\kappa_{\text{bik}}$  and  $\kappa_4 = 4\kappa_{\text{bik}}$ , this inequality is precisely (14.2.16). The remaining inequalities then follow directly by defining  $\kappa_7 = \kappa_2\kappa_3$ ,  $\kappa_6 = \kappa_2\kappa_4$ , and  $\kappa_5 = \kappa_1 + \kappa_7$  and substituting (14.2.16) into (14.2.12)–(14.2.14).

To prove the final assertion it suffices to return to the inequality (14.2.30) and to bound the right-hand side by  $\omega_k + \eta_k$ ; the rate of convergence of  $\{x_k\}$  is inherited by  $\{y_k^{\text{LS}}\}$  because of (14.2.13).  $\square$

## Notes and References for Section 14.2

The quadratic penalty function was originally due to Courant (1943) and was made famous by Fiacco and McCormick (1968). The augmented Lagrangian function was suggested by Powell (1969) and Hestenes (1969). The exponential and other nonquadratic penalty functions are considered by Bertsekas (1982a).

Lemma 14.2.2 is in the spirit of Proposition 2.3 of Bertsekas (1982a, p. 100) but does not require that the Lagrange multiplier estimates stay bounded.

## 14.3 Quadratic Penalty-Function Methods

In this section we shall only be concerned with (14.2.2) when  $y$  is a given *fixed* vector.<sup>235</sup> The consequences of varying  $y$  will be considered in Section 14.4.

It is worth considering (14.2.14) in detail, especially in view of part (i) of Lemma 14.2.2. The first and third terms on the right-hand side of (14.2.14) converge to zero as  $k \in \mathcal{K}$  increases. Thus (14.2.15), and the subsequent first-order criticality of  $x_*$  will follow so long as the middle term  $\mu_k \|y_k - y_*\|$  can be forced to zero. When, as we assume here,  $y_k = y$ , convergence will thus result so long as  $\mu_k$  converges to zero as  $k \in \mathcal{K}$  increases. This then leads us directly to our first algorithm for solving (14.1.2).

---

<sup>235</sup>As we have already mentioned, the more usual quadratic penalty function corresponds to the particular case when  $y = 0$ .

**Algorithm 14.3.1: Quadratic penalty-function algorithm**

**Step 0: Initialization.** An initial point  $x_0^S$ , an initial penalty parameter  $\mu_0 > 0$ , and a stopping tolerance  $\omega_0 \geq 0$  are given, as well as a fixed vector  $y$ . Set  $k = 0$ .

**Step 1: Inner minimization.** (Approximately) solve the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \Phi_y(x, \mu_k) \quad (14.3.1)$$

by applying an unconstrained minimization (trust-region) algorithm, starting from  $x_k^S$  and stopping at an (approximate) solution  $x_k$  for which

$$\|\nabla_x \phi_y(x_k, \mu_k)\| \leq \omega_k. \quad (14.3.2)$$

Set  $y_k = y_y^F(x_k, \mu_k)$ .

**Step 2: Update the penalty parameter.** Choose  $\mu_{k+1} > 0$  and  $\omega_{k+1}$  in such a way as to ensure that

$$\lim_{k \rightarrow \infty} \mu_k = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \omega_k = 0.$$

**Step 3: Choose the next starting point.** Choose a suitable starting point  $x_{k+1}^S$  for the next inner minimization. Increment  $k$  by 1 and return to Step 1.

We then have an immediate corollary of Lemma 14.2.2.

**Theorem 14.3.1** Suppose that the iterates generated by Algorithm 14.3.1 satisfy AI.1 and that the subsequence  $\{x_k\}, k \in \mathcal{K}$ , converge to the point  $x_*$  for which AO.1c holds. Suppose further that AW.1 holds, and let  $y_* = y^{LS}(x_*)$ . Then

- (i)  $x_*$  is a first-order critical point for problem (14.1.2) and the corresponding sequences  $\{y_k\}$  and  $\{y^{LS}(x_k)\}, k \in \mathcal{K}$ , converge to the Lagrange multipliers  $y_*$ ; and
- (ii) there are positive constants  $\kappa_1, \kappa_2$ , and an integer  $k_0$  such that

$$\begin{aligned} \|y_k - y_*\| &\leq \kappa_1 \omega_k + \kappa_2 \|x_k - x_*\|, \\ \|y^{LS}(x_k) - y_*\| &\leq \kappa_2 \|x_k - x_*\|, \end{aligned} \quad (14.3.3)$$

and

$$\|c(x_k)\| \leq \kappa_1 \omega_k \mu_k + \mu_k \|y - y_*\| + \kappa_2 \mu_k \|x_k - x_*\| \quad (14.3.4)$$

for all  $k \geq k_0$ , ( $k \in \mathcal{K}$ ).

Suppose that AO.3b and AW.1b replace AO.1c and AW.1, respectively. Then

(iii) there are constants  $\kappa_3, \kappa_4, \kappa_5, \kappa_6, \kappa_7$ , and  $k_1$  such that

$$\begin{aligned}\|x_k - x_*\| &\leq \kappa_3 \omega_k + \kappa_4 \mu_k, \\ \|y_k - y_*\| &\leq \kappa_5 \omega_k + \kappa_6 \mu_k, \\ \|y^{\text{LS}}(x_k) - y_*\| &\leq \kappa_7 \omega_k + \kappa_6 \mu_k,\end{aligned}\tag{14.3.5}$$

and

$$\|c(x_k)\| \leq \kappa_5 \mu_k \omega_k + \kappa_6 \mu_k^2 + \mu_k \|y - y_*\| \tag{14.3.6}$$

for all  $k \geq k_1 \in \mathcal{K}$ .

**Proof.** Lemma 14.2.2 (i) and the fact that  $y_k = y$  immediately gives the estimates (14.3.3), (14.3.4). But then (14.3.4) and the fact that  $\mu_k$ ,  $\omega_k$ , and  $\|x_k - x_*\|$  all converge to zero while  $\|y - y_*\|$  is a constant show that  $c(x_*) = 0$ . Assertion (i) of the theorem then follows directly from Lemma 14.2.2 (ii). Assertion (iii) follows directly from Lemma 14.2.2 (iii) since the required condition  $\mu_k \leq \mu_{\max}$  is satisfied for all  $k$  sufficiently large, and the index  $k_1$  may merely be increased to reflect this. The constants  $\kappa_4$  and  $\kappa_6$  in (14.3.5), (14.3.6) are those of Lemma 14.2.2 multiplied by  $\|y - y_*\|$ .  $\square$

Theorem 14.3.1 indicates that the R-rate of convergence of the method is normally determined by the rate at which the sequences  $\{\mu_k\}$  and  $\{\omega_k\}$  approach zero. In practice,  $\omega_k$  is usually chosen to be of the same order as  $\mu_k$ , and thus it is the penalty parameter alone that determines the rate of convergence. To simplify the remaining discussion, we shall therefore assume that

$$\omega_k = \omega \mu_k,$$

where  $\omega > 0$  is a given constant.

Choosing the best way to reduce  $\mu_k$  is less obvious. The traditional way is merely to reduce  $\mu_k$  by a constant factor at each stage, that is, to pick

$$\mu_{k+1} = \tau \mu_k \tag{14.3.7}$$

for some  $\tau < 1$ . This choice leads to an R-linear rate of convergence, and a choice of  $\tau = 0.1$  is often seen. A more adventurous strategy is to pick

$$\mu_{k+1} = o(\mu_k), \tag{14.3.8}$$

which leads to an R-superlinear rate. As we shall see, either of these choices can lead to difficulties unless care is taken, which is perhaps why early methods used (14.3.7) with  $\tau$  closer to 1.

Thus far, the quadratic penalty method seems ideal, a sequence of approximate minimizers of a differentiable penalty function leading to a critical point of the underlying constrained optimization problem. But this ignores the difficulty of each unconstrained minimization. A hint of lurking perils is revealed when we consider the Hessian of the penalty function. It is simple to show that

$$\nabla_{xx}\Phi_y(x, \mu) = \nabla_{xx}\ell(x, y^F_y(x, \mu)) + \frac{1}{\mu}A^T(x)A(x). \quad (14.3.9)$$

In view of Theorem 14.3.1, the first term in this expression can be expected to converge to the Hessian of the Lagrangian  $\nabla_{xx}\ell(x_*, y_*)$  as  $\mu$  converges to zero. However, under the same circumstances, the second term will diverge and the entire matrix will asymptotically resemble the rank-deficient matrix  $\frac{1}{\mu}A^T(x_*)A(x_*)$  as  $\mu$  approaches zero.<sup>236</sup> In fact, we have the following precise estimates for all of the eigenvalues of  $\nabla_{xx}\Phi_y(x, \mu)$ .

**Lemma 14.3.2** Suppose that AW.1 holds and that  $x$  lies in a closed, bounded domain  $\Omega$  throughout which  $A(x)$  is of full row rank. Let  $\lambda_{\min}(x, y, \mu)$  and  $\lambda_{\max}(x, y, \mu)$  be the left- and rightmost eigenvalues of the matrix  $\nabla_{xx}\Phi_y(x, \mu)$ . Then, as  $\mu$  approaches zero from above,

(i) if  $m < n$ ,

$$\lambda_{\min}(x, y, \mu) = \lambda_{\min}(x, y) + o(1),$$

where  $\lambda_{\min}(x, y)$  is the smallest eigenvalue of the matrix  $N^T(x)\nabla_{xx}\ell(x, y)$   $N(x)$  and where  $N(x)$  is any matrix satisfying  $A(x)N(x) = 0$  and  $N^T(x)N(x) = I$ ;

(ii) if  $m = n$ ,

$$\lambda_{\min}(x, y, \mu) = \lambda_{\min}(x)/\mu + o(1/\mu),$$

where  $\lambda_{\min}(x)$  is the smallest eigenvalue of the matrix  $A^T(x)A(x)$ ;

(iii) if  $m > 0$ ,

$$\lambda_{\max}(x, y, \mu) = \lambda_{\max}(x)/\mu + o(1/\mu),$$

where  $\lambda_{\max}(x)$  is the largest eigenvalue of the matrix  $A^T(x)A(x)$ ; and

(iv) if  $m = 0$ ,

$$\lambda_{\max}(x, y, \mu) = \lambda_{\max}(x),$$

where  $\lambda_{\max}(x)$  is the largest eigenvalue of the matrix  $\nabla_{xx}f(x)$ .

**Proof.** Let  $Q(x) = (R(x), N(x))$ , where the columns of  $R(x)$  and  $N(x)$  are or-

---

<sup>236</sup>The only exceptions to this are when there are no constraints, or when there are  $n$  constraints. In the former case, the difficult term is absent, while in the latter, the matrix is of full rank. We shall exclude such special cases from our discussion.

thonormal bases for the range- and null-spaces of  $A(x)$ . Since  $Q(x)$  is orthonormal,

$$Q(x)^{-1} \nabla_{xx} \Phi_y(x, \mu) Q(x) = Q^T(x) \nabla_{xx} \Phi_y(x, \mu) Q(x)$$

is a similarity transformation of  $\nabla_{xx} \Phi_y(x, \mu)$  and thus has the same eigenvalues. But

$$\begin{aligned} Q^T(x) \nabla_{xx} \Phi_y(x, \mu) Q(x) &= \\ \begin{pmatrix} R^T(x) \nabla_{xx} \ell(x, y) R(x) + \frac{1}{\mu} (A(x) R(x))^T A(x) R(x) & R^T(x) \nabla_{xx} \ell(x, y) N(x) \\ N^T(x) \nabla_{xx} \ell(x, y) R(x) & N^T(x) \nabla_{xx} \ell(x, y) N(x) \end{pmatrix}. \end{aligned} \quad (14.3.10)$$

As  $\mu$  approaches zero, the  $m$  eigenvectors corresponding to the rightmost eigenvalues of (14.3.10) approach those of

$$\begin{pmatrix} \frac{1}{\mu} (A(x) R(x))^T A(x) R(x) & 0 \\ 0 & 0 \end{pmatrix}, \quad (14.3.11)$$

while, as the remainder must lie in a subspace orthonormal to those of (14.3.11), they have to converge to those of

$$\begin{pmatrix} 0 & 0 \\ 0 & N^T(x) \nabla_{xx} \ell(x, y) N(x) \end{pmatrix}. \quad (14.3.12)$$

The nonzero eigenvalues of (14.3.11) are those of  $1/\mu A^T(x) A(x)$ , while those of (14.3.12) are those of  $N^T(x) \nabla_{xx} \ell(x, y) N(x)$ .  $\square$

In principle, this then indicates that the Hessian of the penalty function will become increasingly ill-conditioned as  $\mu$  decreases, and the unconstrained minimizations successively harder. We illustrate the effect of decreasing  $\mu$  on the contours of  $\Phi_y$  in Figure 14.3.1.

This perceived defect has had the result that most practitioners shun quadratic penalty methods and have long sought other alternatives. However, such a course is unnecessary so long as sufficient care is taken. There are two reasons for such an optimistic view.

Firstly, it is often argued that because the Hessian (14.3.9) may be ill-conditioned, then the solution of linear systems, such as the Newton equations, that involve it may be very inaccurate even if a numerically stable method is used (see Section 4.3.1). However, in this case the ill-conditioning in such systems is very special, and it is easy to transform the system to an equivalent one for which accurate solutions are possible.

To see this, suppose we wish to solve a system of the form

$$\left( M + \frac{1}{\mu} A^T A \right) s = b + \frac{1}{\mu} A^T c; \quad (14.3.13)$$

such a system might be the Newton equations, in which case  $M$  would be the Hessian of the Lagrangian, or might arise as the preconditioning step in an iterative method where the preconditioner is chosen to retain some of the structure (the term  $1/\mu A^T A$ )

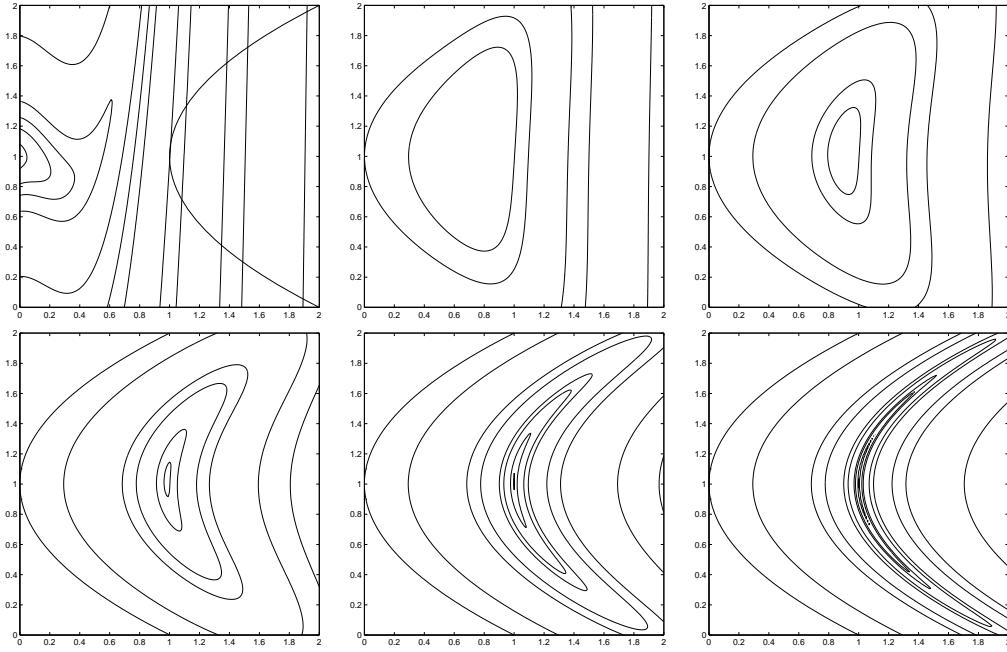


Figure 14.3.1: The two-dimensional example, minimize  $f(x_1, x_2) = 30(4(2x_1 - 1)x_1^2 - x_2 + 1)^2 + (1 + 2x_1)^2$  subject to the constraint  $x_1 - (x_2 - 1)^2 - 1 = 0$ . The top left figure is a contour plot of the problem, including the constraint boundary. The remaining figures illustrate the contours of the quadratic penalty function with  $y = 0$  and  $\mu = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}$ , respectively.

of the original, but where  $M$  is a “simpler” approximation to  $\nabla_{xx}\ell$ . Then, on defining  $t = (As - c)/\mu$ , (14.3.13) may be rewritten as

$$\begin{pmatrix} M & A^T \\ A & -\mu I \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}. \quad (14.3.14)$$

But now, when  $\mu$  is small, the coefficient matrix in (14.3.14) is a small perturbation of the symmetric indefinite matrix

$$\begin{pmatrix} M & A^T \\ A & 0 \end{pmatrix},$$

which is often well-conditioned. Thus, so long as one of the stable methods described in Section 4.3.4 is employed,  $s$  can usually be obtained very accurately.

Secondly, while it is certainly true in a global sense that  $\Phi_y(x, \mu)$  gets harder to minimize as  $\mu$  gets smaller, this ignores the possibility that we may have a very good starting point for the minimization. That is, we may not need to be concerned about how nasty the function is throughout  $\Omega$  since we intend to start close to the minimizer. Early versions of Algorithm 14.3.1 chose the point  $x_k$  ( $k \geq 0$ ) to start each iteration.

However, it is easy to show that this may not be a good choice. For, by definition,

$$\nabla_x \Phi_y(x_k, \mu_{k+1}) = \nabla_x \Phi_y(x_k, \mu_k) + \left(1 - \frac{\mu_k}{\mu_{k+1}}\right) A_k^T(y - y_y^F(x_k, \mu_k)). \quad (14.3.15)$$

Thus, while  $\nabla_x \Phi(x_k, \mu_k)$  becomes progressively smaller, by virtue of (14.3.2), the same is not true for  $\nabla_x \Phi_y(x_k, \mu_{k+1})$ , since the remaining term in (14.3.15) is of order  $\mu_k/\mu_{k+1}$  unless the penalty parameter is decreased very gradually. If, for instance, the update (14.3.7) is used, the left-hand side of (14.3.15) will approach the asymptotic value  $(1 - 1/\tau)A^T(x_*)(y - y_*)$  as  $\mu_k$  approaches zero, and this will be large unless  $\tau$  is very close to 1. If the update (14.3.8) is used, the norm of the gradient actually diverges as  $\mu_k$  converges. Much of this behaviour may be directly attributed to the fact that the multiplier estimate  $y_y^F(x_k, \mu_{k+1})$  at the starting point is poor, for it follows from (14.2.5) and the choice  $y_k = y_y^F(x_k, \mu_k)$  that

$$y_y^F(x_k, \mu_{k+1}) = \frac{\mu_k}{\mu_{k+1}} y_k + \left(1 - \frac{\mu_k}{\mu_{k+1}}\right) y,$$

which, unlike  $y_k$ , does not converge to  $y_*$  unless  $\lim_{k \rightarrow \infty} \mu_k/\mu_{k+1} = 1$ . Thus,  $x_k$  does not immediately appear to be a good starting point.

Fortunately (14.3.15) is deceptive, and a good starting point is actually close at hand. All that is required is that a Newton correction  $s_k$  is made to  $x_k$  for the new penalty parameter  $\mu_{k+1}$  *without* worrying whether the penalty function decreases or not; thus, for this single step, no trust-region (or linesearch) restriction is imposed.<sup>237</sup> As we shall now show, the resulting value  $x_k + s_k$  really is a good starting point for the subsequent minimization.

What is the purpose of this Newton step? And a Newton step for what? Quite simply, the minimization in Step 1 of Algorithm 14.3.1 is aiming to find a root  $(x, y_y^F)$  of the nonlinear system

$$\begin{pmatrix} \nabla_x f(x) - A^T(x)y_y^F \\ c(x) - \mu_{k+1}(y - y_y^F) \end{pmatrix} = 0; \quad (14.3.16)$$

to see this, consider (14.2.4) and (14.2.6). Thus,  $x_k + s_k$  provides a better approximation to the root than  $x_k$ . The Newton correction  $(s, s^{y^F})$  to the estimate  $(x, y_y^F)$  of a root of (14.3.16) satisfies the linear system

$$\begin{pmatrix} \nabla_{xx} \ell(x, y_y^F) & A^T(x) \\ A(x) & -\mu_{k+1} I \end{pmatrix} \begin{pmatrix} s \\ -s^{y^F} \end{pmatrix} = -\begin{pmatrix} \nabla_x f(x) - A^T(x)y_y^F \\ c(x) - \mu_{k+1}(y - y_y^F) \end{pmatrix}$$

or, on eliminating the component  $s^{y^F}$  and using the identities (14.2.6) and (14.2.7),

$$\begin{aligned} \left(\nabla_{xx} \ell(x, y_y^F) + \frac{1}{\mu_{k+1}} A^T(x)A(x)\right) s &= -(\nabla_x f(x) - A^T(x)y_y^F(x, \mu_{k+1})) \\ &= -\nabla_y \Phi(x, \mu_{k+1}). \end{aligned} \quad (14.3.17)$$

---

<sup>237</sup>Since we are essentially just starting a new minimization in a sequence of such problems, an initial increase is not an overriding concern.

It is worth remarking here that although (14.3.17) bears a close resemblance to the Newton equations

$$\nabla_{xx}\Phi_y(x, \mu_{k+1})s = -\nabla_x\Phi_y(x, \mu_{k+1})$$

for the unconstrained minimization of  $\Phi_y(x, \mu_{k+1})$ , the identity (14.3.9) reveals that the two are only the same for the particular choice  $x = x_k$  and  $y_y^F = y_y^F(x_k, \mu_{k+1})$ , a choice we have already rejected in view of (14.3.15).

When  $x = x_k$  and  $y_y^F = y_k$ , we have that

$$\begin{pmatrix} \nabla_x f(x_k) - A^T(x_k)y_k \\ c(x_k) - \mu_{k+1}(y - y_k) \end{pmatrix} = \begin{pmatrix} \nabla_x\Phi_y(x_k, \mu_k) \\ (\mu_k - \mu_{k+1})(y - y_k) \end{pmatrix}. \quad (14.3.18)$$

But the terms on the right-hand side of (14.3.18) may be bounded in norm by  $\kappa_s \mu_k$  for some constant  $\kappa_s \leq 2 \max[\omega, \|y - y_*\|]$ , as  $(x_k, y_k)$  satisfy (14.3.2) and  $y_k$  converges to  $y_*$ . Since the value of the residual to the system (14.3.16) is small at  $(x_k, y_k)$ —unlike at  $(x_k, y_y^F(x_k, \mu_{k+1}))$ —a Newton step for this system is an obvious means of improvement. The Newton correction  $(s_k, s_k^y)$  satisfies the linear system

$$\begin{pmatrix} \nabla_{xx}\ell(x_k, y_k) & A^T(x_k) \\ A(x_k) & -\mu_{k+1}I \end{pmatrix} \begin{pmatrix} s_k \\ -s_k^y \end{pmatrix} = -\begin{pmatrix} \nabla_x f(x_k) - A^T(x_k)y_k \\ c(x_k) - \mu_{k+1}(y - y_k) \end{pmatrix}. \quad (14.3.19)$$

The coefficient matrix of the system (14.3.19) is simply  $K(x_k, y_k, \mu_{k+1})$ , which is a perturbation of  $K(x_*, y_*, 0)$ ; in what follows, we shall assume that AO.3b holds. As (14.3.19) is Newton's method and the initial residual is bounded in norm by  $\kappa_s \mu_k$ , which converges to zero as  $k$  tends to infinity, we thus expect in view of Theorems 3.3.1 and 3.3.2 (p. 52) that the norm of the residual at

$$(x_{k+1}^S, y_{k+1}^S) \stackrel{\text{def}}{=} (x_k + s_k, y_k + s_k^y)$$

will satisfy

$$\left\| \begin{pmatrix} \nabla_x f(x_{k+1}^S) - A^T(x_{k+1}^S)y_{k+1}^S \\ c(x_{k+1}^S) - \mu_{k+1}(y - y_{k+1}^S) \end{pmatrix} \right\| \leq \kappa_9 \mu_k^2 \quad (14.3.20)$$

for some constant  $\kappa_9$ . But the second block of (14.3.20) then implies that

$$\|y_y^F(x_{k+1}^S, \mu_{k+1}) - y_{k+1}^S\| = \|y - y_{k+1}^S - c(x_{k+1}^S)/\mu_{k+1}\| \leq \kappa_9 \frac{\mu_k^2}{\mu_{k+1}}, \quad (14.3.21)$$

and hence the first block of (14.3.20) gives

$$\|\nabla_x\Phi_y(x_{k+1}^S, \mu_{k+1})\| = \|\nabla_x f(x_{k+1}^S) - A^T(x_{k+1}^S)y_y^F(x_{k+1}^S, \mu_{k+1})\| \leq 2\kappa_9 \frac{\mu_k^2}{\mu_{k+1}} \quad (14.3.22)$$

provided that  $\mu_{k+1} \leq 1$ . Now, compare (14.3.15) with (14.3.22). As we have seen, the norm of the former will attain an asymptotic value that is  $\Theta(\mu_k/\mu_{k+1})$ , while the latter is  $O(\mu_k^2/\mu_{k+1})$ . The former is therefore bounded away from zero, while the latter converges to zero so long as  $\mu_k^2 = o(\mu_{k+1})$ . Thus  $x_{k+1}^S$  is likely a better starting value for the minimization of  $\Phi_y(x, \mu_{k+1})$  than  $x_k$ , since its gradient is already small. We also

note that as  $y_k$  converges to  $y_*$  and  $s_k^y$  converges to zero (this follows from (14.3.18), (14.3.19), and AO.3b) (14.3.21) implies that  $y_y^F(x_{k+1}^S, \mu_{k+1})$  also converges to  $y_*$ .

We can take this discussion one stage further. Given the starting point  $x_{k+1}^S$ , an obvious means of finding  $x_{k+1}$  is to apply Newton's method to the minimization of  $\Phi_y(x, \mu_{k+1})$ , particularly since (14.3.22) shows that the gradient of  $\Phi_y(x, \mu_{k+1})$  is small at  $x_{k+1}^S$ . Of course, this needs to be embedded within a trust-region (or linesearch) method to ensure convergence, and the Newton systems are unlikely to be solved accurately in practice, but let us consider what happens if a pure Newton iteration is possible. We let

$$y_{k+1}^F = y_y^F(x_{k+1}^S, \mu_{k+1})$$

and recall from the previous paragraph that  $y_{k+1}^F$  converges to  $y_*$ . The Newton correction  $s_{k+1}^S$  is chosen to satisfy the linear system

$$\nabla_{xx}\Phi_y(x_{k+1}^S, \mu_{k+1})s_{k+1}^S = -\nabla_x\Phi_y(x_{k+1}^S, \mu_{k+1}) \quad (14.3.23)$$

or, equivalently,

$$\begin{aligned} & \left( \nabla_{xx}\ell(x_{k+1}^S, y_{k+1}^F) + \frac{1}{\mu_{k+1}}A^T(x_{k+1}^S)A(x_{k+1}^S) \right) s_{k+1}^S \\ &= -(g(x_{k+1}^S) - A^T(x_{k+1}^S)y_{k+1}^F). \end{aligned} \quad (14.3.24)$$

Now if we define

$$s_{k+1}^{y^F} = y_{k+1}^S - y_{k+1}^F + \frac{1}{\mu_{k+1}}A(x_{k+1}^S)s_{k+1}^S$$

it is straightforward to show that (14.3.24) is equivalent to

$$\begin{aligned} & \begin{pmatrix} \nabla_{xx}\ell(x_{k+1}^S, y_{k+1}^F)A^T(x_{k+1}^S) \\ A(x_{k+1}^S) - \mu_{k+1}I \end{pmatrix} \begin{pmatrix} s_{k+1}^S \\ s_{k+1}^{y^F} \end{pmatrix} \\ &= -\begin{pmatrix} g(x_{k+1}^S) - A^T(x_{k+1}^S)y_{k+1}^S \\ \mu_{k+1}(y_{k+1}^S - y_{k+1}^F) \end{pmatrix} \\ &= -\begin{pmatrix} g(x_{k+1}^S) - A^T(x_{k+1}^S)y_{k+1}^S \\ c(x_{k+1}^S) - \mu_{k+1}(y - y_{k+1}^S) \end{pmatrix}. \end{aligned} \quad (14.3.25)$$

It then follows directly from (14.3.20) that the right-hand side of (14.3.25) is  $O(\mu_k^2)$ . Since  $(x_{k+1}^S, y_{k+1}^F, \mu_{k+1})$  converges to  $(x_*, y_*, 0)$  as  $k \in \mathcal{K}$  tends to infinity, this shows that

$$\left\| \begin{pmatrix} s_{k+1}^S \\ s_{k+1}^{y^F} \end{pmatrix} \right\| \leq \kappa_{10}\mu_k^2 \quad (14.3.26)$$

for some  $\kappa_{10}$ , under AO.3b. Finally, if AW.1b holds,  $\nabla_{xx}\Phi_y(x, \mu_{k+1})$  is Lipschitz continuous throughout  $\Omega$ , with a global Lipschitz constant  $\gamma/\mu_{k+1}$  for some positive  $\gamma$ . Hence it follows from Theorem 3.1.6 (p. 29), (14.3.23), and (14.3.26) that

$$\|\nabla_x\Phi_y(x_{k+1}^S + s_{k+1}^S, \mu_{k+1})\| \leq \frac{1}{2}\gamma\kappa_{10}^2 \frac{\mu_k^4}{\mu_{k+1}}.$$

But this is sufficient to satisfy the termination test (14.3.2) so long as

$$\frac{1}{2}\gamma\kappa_{10}^2 \frac{\mu_k^4}{\mu_{k+1}} \leq \omega\mu_{k+1},$$

which will be asymptotically true so long as

$$\mu_k^2 = o(\mu_{k+1}). \quad (14.3.27)$$

To conclude, if a Newton step is taken from  $x_{k+1}^s$ , and if this step is acceptable for the underlying trust-region (or linesearch) method, then the resulting point  $x_{k+1}^s + s_{k+1}^s$  will be a suitable terminating point for the approximate minimization of  $\Phi_y(x, \mu_{k+1})$  for all  $k$  sufficiently large, so long as the penalty parameter is decreased at most Q-superlinearly.

It remains to show that  $x_{k+1}^s + s_{k+1}^s$  will be a successful step in a typical trust-region method, for example, Algorithm BTR (p. 116). Thus we must show that

$$\|s_{k+1}^s\| \leq \Delta_0^{k+1} \quad (14.3.28)$$

and that

$$\Phi_y(x_{k+1}^s + s_{k+1}^s, \mu_{k+1}) - \Phi_y(x_{k+1}^s, \mu_{k+1}) \leq \frac{1}{2}\eta_1 \langle s, \nabla_x \Phi_y(x_{k+1}^s, \mu_{k+1}) \rangle, \quad (14.3.29)$$

where  $\Delta_0^{k+1}$  is the initial trust-region radius for the approximate solution of the  $(k+1)$ st subproblem (14.3.1) and  $\eta_1$  is the given constant in Algorithm BTR (p. 116). Note that we are assuming here that the model used is a second-order Taylor approximation

$$\begin{aligned} m(x_{k+1}^s + s) &= \Phi_y(x_{k+1}^s, \mu_{k+1}) + \langle s, \nabla_x \Phi_y(x_{k+1}^s, \mu_{k+1}) \rangle \\ &\quad + \frac{1}{2} \langle s, \nabla_{xx} \Phi_y(x_{k+1}^s, \mu_{k+1}) s \rangle \end{aligned} \quad (14.3.30)$$

and, in view of (14.3.23), that

$$m(x_{k+1}^s) - m(x_{k+1}^s + s) = -\frac{1}{2} \langle s, \nabla_x \Phi_y(x_{k+1}^s, \mu_{k+1}) \rangle.$$

The requirement (14.3.28) is easy to ensure since (14.3.26) guarantees that  $s_{k+1}^s$  converges to zero as  $k$  increases. It also follows, as before, that  $\nabla_{xx} \Phi_y(x, \mu_{k+1})$  is Lipschitz continuous throughout  $\Omega$ , with a global Lipschitz constant  $\gamma/\mu_{k+1}$  for some positive  $\gamma$  under AW.1b. Thus Theorem 3.1.5 (p. 29) and (14.3.26) ensure that

$$\begin{aligned} \Phi_y(x_{k+1}^s + s_{k+1}^s, \mu_{k+1}) - \Phi_y(x_{k+1}^s, \mu_{k+1}) &- \frac{1}{2}\eta_1 \langle s, \nabla_x \Phi_y(x_{k+1}^{s \times s}, \mu_{k+1}) \rangle \\ &\leq \frac{1}{2}(1 - \eta_1) \langle s, \nabla_x \Phi_y(x_{k+1}^s, \mu_{k+1}) \rangle + \frac{1}{6}\gamma \|s_{k+1}^s\|^2 \kappa_{10} \frac{\mu_k^2}{\mu_{k+1}}. \end{aligned} \quad (14.3.31)$$

In order to show that (14.3.31) leads to (14.3.29), we need to make one further assumption, namely, that

**AO.3c** the augmented matrix  $K(x_*, y^{\text{LS}}(x_*), 0)$  has inertia  $(n, m, 0)$  for all  $x_* \in \mathcal{L}_*$ .

Remember that this is equivalent to requiring that  $N^T(x_*)\nabla_{xx}\ell(x_*, y^{\text{LS}}(x_*))N(x_*)$  is positive definite (see the notes at the end of Section 3.2.2), and thus by continuity and Lemma 14.3.2

$$\lambda_{\min}(x_{k+1}^S, y, \mu_{k+1}) \geq \frac{1}{2}\lambda_{\min}(x_*, y) > 0 \quad (14.3.32)$$

for all  $k \in \mathcal{K}$  sufficiently large.<sup>238</sup> But the Rayleigh quotient inequality (2.2.5) (p. 19), (14.3.23), and (14.3.32) then imply that

$$\begin{aligned} \langle s_{k+1}^S, \nabla_x \Phi_y(x_{k+1}^S, \mu_{k+1}) \rangle &= -\langle s_{k+1}^S, \nabla_{xx} \Phi_y(x_{k+1}^S, \mu_{k+1}) s_{k+1}^S \rangle \\ &\leq -\lambda_{\min}(x_{k+1}^S, y, \mu_{k+1}) \|s_{k+1}^S\|^2 \\ &\leq -\frac{1}{2}\lambda_{\min}(x_*, y) \|s_{k+1}^S\|^2 < 0. \end{aligned}$$

Combining this with (14.3.31) thus reveals that

$$\begin{aligned} \Phi_y(x_{k+1}^S + s_{k+1}^S, \mu_{k+1}) - \Phi_y(x_{k+1}^S, \mu_{k+1}) - \frac{1}{2}\eta_1 \langle s, \nabla_x \Phi_y(x_{k+1}^{S \times S}, \mu_{k+1}) \rangle \\ \leq -\frac{1}{4}(1-\eta_1)\lambda_{\min}(x_*, y) \|s_{k+1}^S\|^2 + \frac{1}{6}\gamma \|s_{k+1}^S\|^2 \kappa_{10} \frac{\mu_k^2}{\mu_{k+1}}. \end{aligned} \quad (14.3.33)$$

But this then ensures (14.3.29) so long as (14.3.27) holds, as then the second term on the right-hand side of (14.3.33) will be dominated by the first as  $k \in \mathcal{K}$  approaches infinity. Thus the new point  $x_{k+1}^S + s_{k+1}^S$  will be accepted by a typical trust-region method.

## Notes and References for Section 14.3

The prototypical quadratic penalty function method is SUMT (sequential unconstrained minimization techniques) of Fiacco and McCormick (1968), who also provided a number of interesting theoretical results concerning penalty trajectories. Stronger results are possible under suitable convexity assumptions. Lemma 14.3.2 and the consequential observations concerning the ill-conditioning of the Hessian of the penalty function are due to Lootsma (1969) and Murray (1971b). Methods for avoiding the effects of this ill-conditioning were proposed by Broyden and Attia (1984, 1988) and Gould (1986). Much of the remaining material is based on results from Gould (1989) and Conn, Gould, and Toint (1991b).

The solution of the trust-region subproblem (specifically, the necessary modifications to Moré and Sorensen's (1983) method we considered in Section 7.3) when the model has the form (14.3.30) has been examined by Coleman and Hempel (1990). The key is to replace the solution of the generic system (14.3.13), which occurs in Step 1a of Algorithm 7.3.4 (p. 193), when  $M = H + \lambda I$ , by the equivalent system (14.3.14). It is also clear that, since the model Hessian is naturally ill-conditioned, using an  $\ell_2$ -norm trust region is ill advised, and a trust region that more properly reflects the scaling inherent in (14.3.9) is to be preferred. The difficulties associated with the ill-conditioning are most keenly observed if an iterative method, such as the conjugate gradient method we considered in Section 5.1, is applied to the minimization of (14.3.30), particularly in view of the bound (5.1.39) (p. 85) and the worsening conditioning implied by Lemma 14.3.2. In this case, preconditioning is essential. Luenberger (1984, Chapter 12) suggests that at the very least the preconditioner should reflect the terms that lead to the

---

<sup>238</sup>The constant  $\frac{1}{2}$  is arbitrary and can be replaced by any value in the interval  $(0, 1)$ .

ill-conditioning and thus proposes a preconditioner of the form  $I + \frac{1}{\mu} A^T(x)A(x)$ . The effects of this and more general preconditioners are considered by Gould (1999a).

The discussion on the rate of convergence given here is a variant of that formalized by Gould (1989) for linesearch methods. Indeed, it is also straightforward to show that, under AC1.c, the method converges to a single limit point and that the rate is asymptotically two-step Q-superlinear for a suitable sequence of penalty parameters. By this we mean that if we consider a step as being a gradient evaluation, then the sequence formed by neglecting every second step converges Q-superlinearly. In particular, here it is the sequence of terminating values  $\{x_k\}$  that converge Q-superlinearly, and as we have indicated, asymptotically there will only be one gradient evaluation (at  $x_{k+1}^S$ ) between these iterates.

It is also clear, in view of Section 6.6, that an appropriate trust-region method may be used to ensure that the terminating point  $x_k$  at the end of Step 1 of Algorithm 14.3.1 not only satisfies (14.3.2), but is such that  $\nabla_{xx}\Phi_y(x_k, \mu_k)$  is positive semidefinite. Since this condition is equivalent to  $K(x_k, y_k, \mu_k)$  having precisely  $m$  negative eigenvalues (see the notes at the end of Section 3.2.2), the same will be true of  $K(x_*, y_*, 0)$ , which implies that  $(x_*, y_*)$  will be a second-order critical point for (14.1.2).

For general problems of the form (14.1.1), almost identical methods based on a sequential minimization of the function

$$\Psi_y(x, \mu) = f(x) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \begin{cases} -c_i(x)y_i + \frac{1}{2\mu} c_i(x)^2 & \text{if } c_i(x) \leq \mu y_i \text{ or } i \in \mathcal{E}, \\ -\frac{1}{2}\mu y_i^2 & \text{otherwise,} \end{cases} \quad (14.3.34)$$

where  $y_{\mathcal{I}} \geq 0$  (see Rockafellar, 1974), are appropriate. Although this function has discontinuous second derivatives, it can be shown that, under a strict complementary slackness assumption, these do not interfere with the ultimate convergence of the method (see Gould, 1989).

## 14.4 Augmented Lagrangian Function Methods

We now turn to the possibility that the vector  $y$  in (14.2.2) may be allowed to vary. As before, it is worth considering the bound

$$\|c(x_k)\| \leq \kappa_1 \omega_k \mu_k + \mu_k \|y_k - y_*\| + \kappa_2 \mu_k \|x_k - x_*\|, \quad (14.4.1)$$

in this light, especially in view of part (i) of Lemma 14.2.2. The first and third terms on the right-hand side of (14.4.1) converge to zero as  $k \in \mathcal{K}$  increases. Thus  $c(x_*) = 0$ , and the subsequent first-order criticality of  $x_*$  will follow so long as the middle term  $\mu_k \|y_k - y_*\|$  can be forced to zero. We saw in the previous section that this is easy to ensure if  $\mu_k$  converges to zero while  $y_k$  stays bounded. But it is equally true if  $\mu_k$  is bounded away from zero, so long as  $y_k$  can be encouraged to converge to  $y_*$ . The advantage of this alternative is simply that the inevitable ill-conditioning of the Hessian of (14.2.2) may be avoided if  $\mu_k$  can be bounded away from zero. Observe, in Figure 14.4.1, how the minimizer of the augmented Lagrangian function moves towards  $x_*$  as  $y$  approaches  $y_*$ .

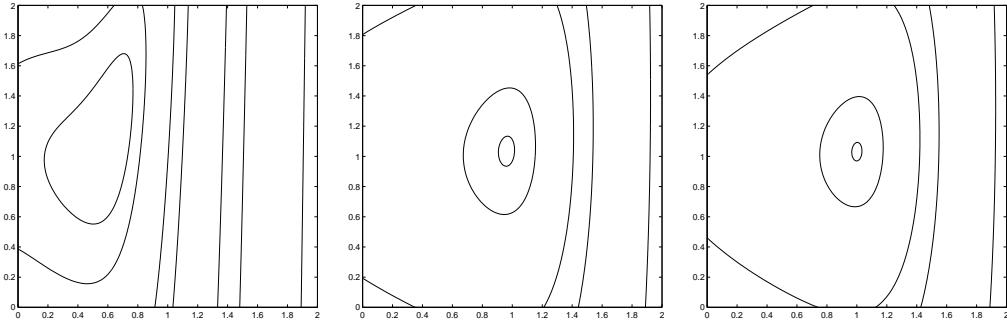


Figure 14.4.1: The augmented Lagrangian function for the two-dimensional example, minimize  $f(x_1, x_2) = 30(4(2x_1 - 1)x_1^2 - x_2 + 1)^2 + (1 + 2x_1)^2$  subject to the constraint  $x_1 - (x_2 - 1)^2 - 1 = 0$ . The figures illustrate, from left to right, the contours of the function with  $\mu = 0.1$  and  $y = 50, 3000, 3852 (= y_*)$ , respectively. (It is interesting to compare this figure with Figure 14.3.1.)

Part (i) of Lemma 14.2.2 also strongly suggests how we might pick  $y_k$ , for the bounds (14.2.12) and (14.2.13) show that both the first-order and least-squares Lagrange multiplier estimates converge to  $y_*$ . We must inject a note of caution here as  $y_k^F$  and  $y_k^{LS}$  are only available once  $x_k$  is known. It does not follow that these values are appropriate estimates for the multipliers at  $x_{k+1}$ , since  $x_{k+1}$  may converge to a different limit. Indeed, it is quite possible that the “obvious” strategy of picking

$$y_{k+1} = y_k^F \quad \text{or} \quad y_{k+1} = y_k^{LS}$$

may lead to divergent multiplier estimates, or at the very least may not ensure that  $\mu_k \|y_k - y_*\|$  converges to zero. Thus, we have to be selective as to when to change  $y$ .

To emphasize that the augmented Lagrangian now also depends on a variable  $y$ , we shall write  $\Phi_y(x, \mu)$  as

$$\Phi(x, y, \mu) = f(x) - \sum_{i \in \mathcal{E}} y_i c_i(x) + \frac{1}{2\mu} \sum_{i \in \mathcal{E}} c_i(x)^2, \quad (14.4.2)$$

where, as before, the components,  $y_i$ , of the vector  $y$  are known as Lagrange multiplier estimates and  $\mu$  is known as the penalty parameter. We also emphasize the dependence of first-order Lagrange multiplier estimates

$$y^F(x, y, \mu) = y - \frac{c(x)}{\mu} \quad (14.4.3)$$

on  $y$ , and, as in (14.2.7), we shall make much use of the identity

$$\nabla_x \Phi(x, y, \mu) = \nabla_x \ell(x, y^F(x, y, \mu)).$$

In order to solve problem (14.1.2), we consider the following basic algorithm, Algorithm 14.4.1.

**Algorithm 14.4.1: Augmented Lagrangian algorithm**

**Step 0: Initialization.** An initial estimate of the solution,  $x_0^s$ , and a vector of Lagrange multiplier estimates,  $y_0$ , are given, along with the least allowable decrease in the penalty parameter,  $\tau < 1$ . Set the initial penalty parameter  $\mu_0 < 1$ , the initial convergence tolerances  $\omega_0 > 0$  and  $\eta_0 > 0$ , and  $k = 0$ .

**Step 1: Inner iteration.** (Approximately) solve the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \Phi(x, y_k, \mu_k) \quad (14.4.4)$$

by applying an unconstrained minimization (trust-region) algorithm, starting from  $x_k^s$  and stopping at an (approximate) solution  $x_k$  for which

$$\|\nabla_x \Phi(x_k, y_k, \mu_k)\| \leq \omega_k. \quad (14.4.5)$$

If

$$\|c(x_k)\| \leq \eta_k, \quad (14.4.6)$$

execute Step 2. Otherwise, execute Step 3.

**Step 2: Update Lagrange multiplier estimates.** Set

$$y_{k+1} = y^F(x_k, y_k, \mu_k) \quad \text{and} \quad \mu_{k+1} \leq \mu_k, \quad (14.4.7)$$

pick positive  $\omega_{k+1}$  and  $\eta_{k+1}$  so that

$$\lim_{k \rightarrow \infty} \omega_k = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \eta_k = 0, \quad (14.4.8)$$

choose the next starting point  $x_{k+1}^s$ , increment  $k$  by 1, and go to Step 1.

**Step 3: Significantly reduce the penalty parameter.** Set

$$y_{k+1} = y_k \quad \text{and} \quad \mu_{k+1} \leq \tau \mu_k, \quad (14.4.9)$$

pick positive  $\omega_{k+1}$  and  $\eta_{k+1}$  so that (14.4.8) holds, choose the next starting point  $x_{k+1}^s$ , increment  $k$  by 1, and go to Step 1.

The motivation for this algorithm is quite straightforward. So long as the constraint violation converges to zero, Lemma 14.2.2 (ii) will ensure convergence even when  $\mu_k$  does not converge to zero. When the algorithm does not appear to be converging to a feasible point, as a last resort we must drive the penalty parameter to zero and ensure that the Lagrange multiplier estimates do not behave too badly. The convergence of such a scheme is guaranteed, since in this case the iteration is essentially that used in the quadratic penalty function method discussed in Section 14.3. The only outstanding

difficulty is how to ensure that the multiplier estimates do not behave badly, and to do this we need to impose further conditions on the convergence tolerances.

Algorithm 14.4.1 is designed specifically for the first-order estimate (14.4.3); the multiplier estimates are encouraged to behave well as a consequence of the test (14.4.6). For large-scale computations, first-order estimates are significantly less expensive to compute than other options, and in this case Algorithm 14.4.1 is directly relevant.

In practice, the algorithm should be halted if, at the start of Step 2,

$$\|\nabla_x \Phi(x_k, y_k, \mu_k)\| \leq \omega_* \quad \text{and} \quad \|c(x_k)\| \leq \eta_*$$

for some appropriate, small positive tolerances  $\omega_*$  and  $\eta_*$ . See Section 17.4.3.3 for further discussions on stopping rules.

To show that the method is globally convergent, it is necessary to show that the Lagrange multiplier estimates cannot grow too fast relative to the decrease in the penalty parameter. To do so, we need to make the following technical assumption about the tolerances  $\eta_i$ .

**AA.9** The sequence  $\{\eta_i\}$  satisfies the bound

$$\sum_{l=1}^{k_{i+1}-k_i} \eta_{k_i+l} \leq \kappa_\eta \mu_{k_i+1}^{\alpha_\eta}$$

for some positive  $\kappa_\eta$  and  $\alpha_\eta$  and all  $i \geq 0$ , where  $\mathcal{K} = \{k_0, k_1, k_2, \dots\}$  is the set of the indices of the iterations at which Step 3 of Algorithm 14.4.1 is executed.

This assumption essentially ensures that the constraint violation is not allowed to grow too large without our making every effort to reduce it using the only means at our disposal, namely, by reducing the penalty parameter. We shall shortly suggest a practical framework for which AA.9 holds. We have the following crucial result.

**Lemma 14.4.1** Suppose that Step 3 of Algorithm 14.4.1 is executed infinitely often and that the tolerances  $\eta_k$  satisfy AA.9. Then the product  $\mu_k \|y_k\|$  converges to zero.

**Proof.** Since Step 3 of the algorithm is executed infinitely often,  $\mu_k$  converges to zero. Let  $\mathcal{K} = \{k_0, k_1, k_2, \dots\}$  be the set of the indices of the iterations in which Step 3 of the algorithm is executed. We consider how the Lagrange multiplier estimates change between two successive iterations indexed in the set  $\mathcal{K}$ . At iteration  $k_i + j$ , for  $k_i < k_i + j \leq k_{i+1}$ , we have

$$y_{k_i+j} = y_{k_i} + \sum_{l=1}^{j-1} \frac{c(x_{k_i+l})}{\mu_{k_i+l}} \quad (14.4.10)$$

and

$$\mu_{k_i+1} = \mu_{k_i+j} = \mu_{k_i+1} \leq \tau \mu_{k_i}, \quad (14.4.11)$$

where the summation in (14.4.10) is null if  $j = 1$ . Now suppose that  $j > 1$ . Then for the set of iterations  $k_i + l, 1 \leq l < j$ , Step 2 of the algorithm must have been executed, and hence we must have that

$$\|c(x_{k_i+l})\| \leq \eta_{k_i+l}. \quad (14.4.12)$$

Combining equations (14.4.10)–(14.4.12) and invoking AA.9, we obtain the bound

$$\|y_{k_i+j}\| \leq \|y_{k_i}\| + \sum_{l=1}^{j-1} \frac{\|c(x_{k_i+l})\|}{\mu_{k_i+l}} \leq \|y_{k_i}\| + \kappa_\eta \frac{\mu_{k_i+1}^{\alpha_\eta}}{\mu_{k_i+j}} \leq \|y_{k_i}\| + \kappa_\eta \tau^{\alpha_\eta} \frac{\mu_{k_i}^{\alpha_\eta}}{\mu_{k_i+j}}.$$

Thus we obtain that

$$\mu_{k_i+j} \|y_{k_i+j}\| \leq \tau \mu_{k_i} \|y_{k_i}\| + \kappa_\eta \tau^{\alpha_\eta} \mu_{k_i}^{\alpha_\eta}. \quad (14.4.13)$$

Equation (14.4.13) is also satisfied when  $j = 1$  as equations (14.4.9) and (14.4.11) give  $\mu_{k_i+1} \|y_{k_i+1}\| \leq \tau \mu_{k_i} \|y_{k_i}\|$ . Hence from (14.4.13),

$$\mu_{k_i+1} \|y_{k_i+1}\| \leq \tau \mu_{k_i} \|y_{k_i}\| + \kappa_\eta \tau^{\alpha_\eta} \mu_{k_i}^{\alpha_\eta}. \quad (14.4.14)$$

Equation (14.4.14) then gives that  $\mu_{k_i} \|y_{k_i}\|$  converges to zero as  $k$  increases, for, if we define

$$\alpha_i = \mu_{k_i} \|y_{k_i}\| \quad \text{and} \quad \beta_i = \kappa_\eta \mu_{k_i}^{\alpha_\eta}, \quad (14.4.15)$$

relations (14.4.11), (14.4.14), and (14.4.15) give that

$$\alpha_{i+1} \leq \tau \alpha_i + \tau^{\alpha_\eta} \beta_i \quad \text{and} \quad \beta_{i+1} \leq \tau^{\alpha_\eta} \beta_i \quad (14.4.16)$$

and hence that

$$0 \leq \alpha_i \leq \tau^i \alpha_0 + \tau^{i\alpha_\eta} \sum_{l=0}^{i-1} \tau^{l(1-\alpha_\eta)} \beta_0. \quad (14.4.17)$$

If  $\alpha_\eta < 1$ , the sum in (14.4.17) can be bounded to give

$$0 \leq \alpha_i \leq \tau^i \alpha_0 + \tau^{i\alpha_\eta} \beta_0 / (1 - \tau^{1-\alpha_\eta}),$$

whereas if  $\alpha_\eta > 1$  we obtain the alternative

$$0 \leq \alpha_i \leq \tau^i (\alpha_0 + \tau^{\alpha_\eta-1} \beta_0 / (1 - \tau^{\alpha_\eta-1})),$$

and if  $\alpha_\eta = 1$ ,

$$0 \leq \alpha_i \leq \tau^i \alpha_0 + i \tau^i \beta_0.$$

But both  $\alpha_0$  and  $\beta_0$  are finite. Thus, as  $i$  increases,  $\alpha_i$  converges to zero; equation (14.4.16) implies that  $\beta_i$  converges to zero. The required result now immediately follows since the right-hand side of (14.4.13) converges to zero.  $\square$

This leads directly to a general global convergence result.

**Theorem 14.4.2** Assume that AW.1 holds. Let  $x_*$  be any limit point of the sequence  $\{x_k\}$  generated by Algorithm 14.4.1 for which AO.1c and AI.1 hold and let  $\mathcal{K}$  be the set of indices of an infinite subsequence of the  $x_k$  whose limit is  $x_*$ . Then conclusions (i)–(iv) of Lemma 14.2.2 hold.

**Proof.** The assumptions given are sufficient to reach the conclusions of part (i) of Lemma 14.2.2. We now show that (14.2.15) holds for Algorithm 14.4.1. To see this, we consider two separate cases.

- (a) If Step 3 is executed infinitely often,  $\mu_k$  converges to zero and Lemma 14.4.1 shows that  $\mu_k \|y_k - y_*\|$  converges to zero. But then, inequality (14.4.1) gives the required result.
- (b) If Step 3 is only executed finitely often, Step 2 must be executed every iteration for  $k$  sufficiently large. But this implies that (14.4.6) is always satisfied ( $k$  large enough) and  $\eta_k$  converges to zero. Hence  $c(x_k)$  converges to zero.

Hence equation (14.2.15) is satisfied and thus conclusions (ii)–(iv) of Lemma 14.2.2 hold.  $\square$

Notice that Theorem 14.4.2 holds regardless of the actual choice of the sequence  $\{\omega_k\}$  provided that it converges to zero. A reasonable set of update rules is given in Algorithm 14.4.2.

#### Algorithm 14.4.2: Tolerance update rules

Let  $\omega, \eta, \alpha_\omega, \alpha_\eta, \beta_\omega$ , and  $\beta_\eta$  be given positive constants. Set the initial values

$$\omega_0 = \omega \mu_0^{\alpha_\omega} \quad \text{and} \quad \eta_0 = \eta \mu_0^{\alpha_\eta}.$$

In Step 2 of the augmented Lagrangian algorithm, pick

$$\omega_{k+1} = \omega_k \mu_{k+1}^{\beta_\omega} \quad \text{and} \quad \eta_{k+1} = \eta_k \mu_{k+1}^{\beta_\eta},$$

while in Step 3 set

$$\omega_{k+1} = \omega \mu_{k+1}^{\alpha_\omega} \quad \text{and} \quad \eta_{k+1} = \eta \mu_{k+1}^{\alpha_\eta}.$$

It is easy to see that this strategy ensures that AA.9 holds, since

$$\sum_{l=1}^{k_{i+1}-k_i} \eta \mu_{k_i+l} = \sum_{l=1}^{k_{i+1}-k_i} \eta \mu_{k_i+1}^{\alpha_\eta + (l-1)\beta_\eta} < \eta \mu_{k_i+1}^{\alpha_\eta} \sum_{l=1}^{\infty} \eta \mu_0^{(l-1)\beta_\eta} = \frac{\eta}{1 - \mu_0^{\beta_\eta}} \mu_{k_i+1}^{\alpha_\eta}$$

as  $\mu_{k+1} \leq \mu_0$ . In order to try to allow the traditional multiplier iteration to take over, the test on the size of the constraints (14.4.6) using these rules is based upon the size that might be expected if the multiplier iteration is converging. Lemma 14.2.2 (iii) suggests that if  $\mu$  is sufficiently small, the first-order multiplier update  $y_{k+1} = y_k^F$  and the value of the constraint violation  $\|c(x_k)\|$  will converge R-linearly whenever  $\omega_k$  does. The particular choice of  $\omega_k$  above is made with this in mind. To recognize when linear convergence has set in, it is necessary to monitor  $\|c(x_k)\|$  to look for linear reduction, and this is what this particular choice of  $\eta_k$  aims for.

Algorithms 14.4.1 and 14.4.2 use a number of free parameters. To give the reader some feel for what might be typical values, we suggest that for well-scaled problems

$$\alpha_\omega = \beta_\omega = \eta = \omega = 1, \quad \alpha_\eta = \mu_0 = 0.1, \quad \beta_\eta = 0.9, \quad \text{and} \quad \tau = 0.01$$

are appropriate.

Thus far, we have not discussed how the penalty parameter should be updated in the algorithm, except for insisting that there is a significant reduction in Step 3. So, first consider (14.4.9). Here, it is traditional to choose  $\mu_{k+1} = \tau \mu_k$ , but, in view of the estimates in (iii) of Lemma 14.2.2, it may be better to pick

$$\mu_{k+1} = \mu_k \min[\tau, \mu_k]$$

so as to encourage a faster R-rate of convergence if the penalty parameter is forced to zero; of course, we must take precautions in this case to mitigate the effects of ill-conditioning. Turning to Step 2, it is traditional merely to choose  $\mu_{k+1} = \mu_k$  in (14.4.7), but again a reduction will likely increase the rate of convergence. However, there is a good reason why we should eventually settle for  $\mu_{k+1} = \mu_k$  in this step, for we now show that the penalty parameter will be bounded away from zero in Algorithm 14.4.1 so long as an appropriate set of rules is used to update the convergence tolerances, and provided slightly stronger assumptions are made about the problem. As we said earlier, this is important, as methods for solving the inner iteration subproblem may encounter difficulties if the penalty parameter converges to zero since this allows the Hessian of the augmented Lagrangian to become increasingly ill-conditioned. We thus assume that

**AA.10** the penalty parameter update in the second part of (14.4.7) in Step 2 of the algorithm is replaced by  $\mu_{k+1} = \mu_k$  for all  $k \geq k_1$  and some  $k_1 \geq 0$ .

With this assumption, we have the following result.

**Theorem 14.4.3** Suppose that the iterates  $\{x_k\}$  generated by Algorithm 14.4.1 under AA.10, with the tolerance updating rule given in Algorithm 14.4.2, converge to the single limit point  $x_*$ ; that AO.3b, AW.1b, and AI.1 hold; and that  $\alpha_\eta$  and  $\beta_\eta$  satisfy

$$\alpha_\eta < \alpha \stackrel{\text{def}}{=} \min[1, \alpha_\omega], \quad (14.4.18)$$

$$\beta_\eta < \min[1, \beta_\omega]. \quad (14.4.19)$$

Then there is a constant  $\mu_* > 0$  such that  $\mu_k \geq \mu_*$  for all  $k$ .

**Proof.** Suppose, otherwise, that is, that  $\mu_k$  tends to zero. Then, Step 3 of the algorithm must be executed infinitely often. We aim to obtain a contradiction to this statement by showing that Step 2 is always executed for  $k$  sufficiently large. We note that our assumptions are sufficient for the conclusions of Theorem 14.4.2 to hold.

Firstly, we show that the sequence of Lagrange multipliers  $\{y_k\}$  converge to  $y_*$ . The result is clear if Step 2 is executed infinitely often, as each time the step is executed,  $y_{k+1} = y_k^F$  and the inequality (14.2.12) guarantees that  $y_k^F$  converges to  $y_*$ . Suppose that Step 2 is not executed infinitely often. Then  $\|y_k - y_*\|$  will remain fixed for all  $k \geq k_0$  for some  $k_0$ , as Step 3 is executed for each remaining iteration. But then (14.4.1) implies that  $\|c(x_k)\| \leq \kappa_{11}\mu_k$  for some constant  $\kappa_{11}$  for all  $k \geq k_2 \geq k_0$ . As  $\mu_k$  converges to zero as  $k$  increases,  $\kappa_{11}\mu_k \leq \eta_0\mu_k^{\alpha_\eta} = \eta_k$  for all  $k$  sufficiently large. But then inequality (14.4.6) must be satisfied for some  $k \geq k_0$ , which is impossible as this would imply that Step 2 is again executed. Hence, Step 2 must be executed infinitely often. Therefore,  $\{y_k\}$  converge to  $y_*$  and  $\mu_k\|y_k - y_*\|$  tends to zero as  $k$ .

Let  $\omega_k$  be as generated by Algorithm 14.4.1. Notice that, by construction,

$$\omega_k \leq \omega\mu_k^{\alpha_\omega}; \quad (14.4.20)$$

this follows by definition if Step 3 of the algorithm occurs and because the penalty parameter is not increased while  $\omega_k$  is reduced when Step 2 occurs. We shall apply Lemma 14.2.2 to the iterates generated by the algorithm; we identify the set  $\mathcal{K}$  with the set of positive integers and the scalars  $\mu_k$  with the set of penalty parameters computed in Steps 2 and 3 of the algorithm. Therefore, we can ensure that  $\mu_k$  is sufficiently small so that Lemma 14.2.2 (iii) applies to Step 1 of Algorithm 14.4.1 and thus that there is an integer  $k_2$  and constants  $\kappa_5$  and  $\kappa_6$  so that (14.2.17) and (14.2.18) hold for all  $k \geq k_2$ . Let  $k_3$  be the smallest integer such that

$$\mu_k \leq \frac{1}{\kappa_6}, \quad (14.4.21)$$

$$\mu_k^{1-\alpha_\eta} \leq \frac{\eta}{\omega(\kappa_5 + 2)}, \quad (14.4.22)$$

and

$$\mu_k^{1-\beta_\eta} \leq \min \left[ \frac{1}{\kappa_{12}}, \frac{\eta}{\omega(\kappa_5 + 2\kappa_{12})} \right] \quad (14.4.23)$$

for all  $k \geq k_3$ , where  $\kappa_{12} = \kappa_5 + \kappa_6$ . Furthermore, let  $k_4$  be such that

$$\|y_k - y_*\| \leq \omega \quad (14.4.24)$$

for all  $k \geq k_4$ . Now define  $k_5 = \max[k_1, k_2, k_3, k_4]$ , where  $k_1$  is as given in AA.10; let

$$\mathcal{S} = \{k \mid \text{Step 3 is executed at iteration } k - 1 \text{ and } k \geq k_5\};$$

and let  $k_0$  be the smallest element of  $\mathcal{S}$ . By assumption,  $\mathcal{S}$  has an infinite number of elements.

For iteration  $k_0$ ,  $\omega_{k_0} = \omega\mu_{k_0}^{\alpha_\omega}$  and  $\eta_{k_0} = \eta\mu_{k_0}^{\alpha_\eta}$ . Then (14.2.18) gives

$$\begin{aligned}
\|c(x_{k_0})\| &\leq (\mu_{k_0} + \kappa_6 \mu_{k_0}^2) \|y_{k_0} - y_*\| + \kappa_5 \omega_{k_0} \mu_{k_0} \\
&\leq 2\mu_{k_0} \|y_{k_0} - y_*\| + \kappa_5 \omega_{k_0} \mu_{k_0} && \text{(from (14.4.21))} \\
&\leq 2\omega \mu_{k_0} + \kappa_5 \omega \mu_{k_0}^{1+\alpha_\omega} && \text{(from (14.4.24))} \quad (14.4.25) \\
&\leq \omega(\kappa_5 + 2)\mu_{k_0} && \text{(as } \mu_{k_0} < 1\text{)} \\
&\leq \eta \mu_{k_0}^{\alpha_\eta} = \eta_{k_0} && \text{(from (14.4.22)).}
\end{aligned}$$

Thus, from (14.4.25), Step 2 of Algorithm 14.4.1 will be executed with  $y_{k_0+1} = y_{k_0}^F$ . Inequality (14.2.17), in conjunction with (14.4.18), (14.4.20), and (14.4.24) guarantee that

$$\|y_{k_0+1} - y_*\| \leq \kappa_5 \omega_{k_0} + \kappa_6 \mu_{k_0} \|y_{k_0} - y_*\| \leq \omega \kappa_{12} \mu_{k_0}^\alpha. \quad (14.4.26)$$

We shall now suppose that Step 2 is executed for iterations  $k_0 + i$  ( $0 \leq i \leq j$ ) and that

$$\|y_{k_0+i+1} - y_*\| \leq \omega \kappa_{12} \mu_{k_0}^{\alpha + \beta_n i}. \quad (14.4.27)$$

Inequalities (14.4.25) and (14.4.26) show that this is true for  $j = 0$ . We aim to show that the same is true for  $i = j + 1$ . Under our supposition, we have, for iteration  $k_0 + j + 1$ , that  $\mu_{k_0+j+1} = \mu_{k_0}$ ,  $\omega_{k_0+j+1} = \omega \mu_{k_0}^{\beta_\omega(j+1)+\alpha_\omega}$ , and  $\eta_{k_0+j+1} = \eta \mu_{k_0}^{\beta_\eta(j+1)+\alpha_\eta}$ . Then (14.2.18) gives

$$\begin{aligned}
\|c(x_{k_0+j+1})\| &\leq (\mu_{k_0+j+1} + \kappa_6 \mu_{k_0+j+1}^2) \|y_{k_0+j+1} - y_*\| + \kappa_5 \omega_{k_0+j+1} \mu_{k_0+j+1} \\
&\leq 2\mu_{k_0+j+1} \|y_{k_0+j+1} - y_*\| + \kappa_5 \omega_{k_0+j+1} \mu_{k_0+j+1} \quad (\text{from (14.4.21)}) \\
&\leq 2\omega_{k_{12}} \mu_{k_0} \mu_{k_0}^{\alpha+\beta\eta j} + \kappa_5 \omega \mu_{k_0}^{\alpha_\omega+\beta_\omega(j+1)+1} \quad (\text{from (14.4.27)}) \\
&\leq (2\omega \kappa_{12} \mu_{k_0} \mu_{k_0}^{\alpha_\eta+\beta\eta j} + \kappa_5 \omega \mu_{k_0}^{\alpha_\eta+\beta\eta(j+1)+1}) \quad (\text{from (14.4.18)}), \\
&\hspace{30em} (14.4.19), \text{ and as } \mu_{k_0} < 1 \\
&\leq \omega(\kappa_5 + 2\kappa_{12}) \mu_{k_0}^{1-\beta\eta} \mu_{k_0}^{\beta\eta(j+1)+\alpha_\eta} \quad (\text{as } \mu_{k_0} < 1) \\
&\leq \eta \mu_{k_0}^{\beta\eta(j+1)+\alpha_\eta} = \eta_{k_0+j+1} \quad (\text{from (14.4.23)}).
\end{aligned}$$

Thus Step 2 of Algorithm 14.4.1 will be executed with  $y_{k_0+j+2} = y_{k_0+j+1}^F$ . Inequality (14.2.17) then guarantees that

$$\begin{aligned}\|y_{k_0+j+2} - y_*\| &\leq \kappa_5 \omega_{k_0+j+1} + \kappa_6 \mu_{k_0+j+1} \|y_{k_0+j+1} - y_*\| \\ &\leq \omega \kappa_5 \mu_{k_0}^{\alpha_\omega + \beta_\omega(j+1)} + \omega \kappa_6 \kappa_{12} \mu_{k_0} \mu_{k_0}^{\alpha + \beta_\eta j} \quad (\text{from (14.4.27)}) \\ &\leq \omega \kappa_5 \mu_{k_0}^{\alpha + \beta_\eta(j+1)} + \omega \kappa_6 \kappa_{12} \mu_{k_0} \mu_{k_0}^{\alpha + \beta_\eta j} \quad (\text{from (14.4.18), (14.4.19), and as } \mu_{k_0} < 1) \\ &= \omega (\kappa_5 + \kappa_6 \kappa_{12} \mu_{k_0}^{1-\beta_\eta}) \mu_{k_0}^{\alpha + \beta_\eta(j+1)} \quad (\text{from (14.4.23)}) \\ &\leq \omega (\kappa_5 + \kappa_6) \mu_{k_0}^{\alpha + \beta_\eta(j+1)} \\ &= \omega \kappa_{12} \mu_{k_0}^{\alpha + \beta_\eta(j+1)},\end{aligned}$$

which establishes (14.4.27) for  $i = j + 1$ . Hence, Step 2 of the algorithm is executed for all iterations  $k \geq k_0$ . But this implies that  $\mathcal{S}$  is finite, which contradicts the assumption that Step 3 is executed infinitely often. Hence the theorem is proved.  $\square$

The assumption that  $\{x_k\}$  has a single limit point is crucial, as examples have been constructed to show that otherwise, if the sequence has two limit points, the penalty parameter may in fact converge to zero.

Algorithm 14.4.1 is quite specifically tied to the first-order Lagrange multiplier estimates  $y^F(x_k, y_k, \mu_k)$ . In practice, other estimates are often available. It is therefore of interest to develop more general algorithms, which are capable of using such estimates. Such a method is given in Algorithm 14.4.3.

#### Algorithm 14.4.3: Alternative augmented Lagrangian algorithm

**Step 0: Initialization.** An initial estimate of the solution,  $x_0^S$ , and vector of Lagrange multiplier estimates,  $y_0$ , are given, as are positive constants  $\tau < 1$ ,  $\gamma < 1$ , and  $\nu$ . Set the initial penalty parameter  $\mu_0 < 1$ , the initial convergence tolerances  $\omega_0 > 0$  and  $\eta_0 > 0$ , and  $k = 0$ .

**Step 1: Inner iteration.** (Approximately) solve the problem (14.4.4) by applying an unconstrained minimization (trust-region) algorithm, starting from  $x_k^S$  and stopping at an (approximate) solution  $x_k$  for which (14.4.5) is satisfied. Compute a new vector of Lagrange multiplier estimates  $y_{k+1}^E$ . If (14.4.6) holds execute Step 2. Otherwise, execute Step 3.

**Step 2: Leave the penalty parameter unchanged.** Set

$$\mu_{k+1} \leq \mu_k \quad \text{and} \quad y_{k+1} = \begin{cases} y_{k+1}^E & \text{if } \|y_{k+1}^E\| \leq \nu(\mu_{k+1})^{-\gamma} + \theta_{k+1}, \\ y_k & \text{otherwise;} \end{cases}$$

pick positive  $\omega_{k+1}$ ,  $\eta_{k+1}$ , and  $\theta_{k+1}$  so that

$$\lim_{k \rightarrow \infty} \omega_k = 0, \quad \lim_{k \rightarrow \infty} \eta_k = 0, \quad \text{and} \quad \lim_{k \rightarrow \infty} \theta_k = 0; \quad (14.4.28)$$

choose the next starting point  $x_{k+1}^S$ , increment  $k$  by 1, and go to Step 1.

**Step 3: Significantly reduce the penalty parameter.** Set

$$\mu_{k+1} \leq \tau\mu_k \quad \text{and} \quad y_{k+1} = \begin{cases} y_{k+1}^E & \text{if } \|y_{k+1}^E\| \leq \nu(\mu_{k+1})^{-\gamma} + \theta_{k+1}, \\ y_k & \text{otherwise;} \end{cases}$$

pick positive  $\omega_{k+1}$ ,  $\eta_{k+1}$ , and  $\theta_{k+1}$  so that (14.4.28) holds; choose the next starting point  $x_{k+1}^S$ , increment  $k$  by 1, and go to Step 1.

This algorithm differs from its predecessor in its use of multiplier update. Algorithm 14.4.3 allows any multiplier estimate to be used. This extra freedom means that tighter control must be maintained on the acceptance of the estimates to make sure that they do not grow unacceptably fast. In this algorithm, we have in mind using any of the well-known Lagrange multiplier update formulae, including the first-order update (14.4.3) (used in Algorithm 14.4.1), the least-squares update (14.2.8), and other updates summarized, for instance, by Tapia (1977). We note, however, that some of these updates may require a significant amount of computation and this may prove prohibitively expensive for large-scale problems. Algorithm 14.4.3 is identical to Algorithm 14.4.1 except for the allowed Lagrange multiplier updates, the fact that these updates may also occur in Step 3, and the presence of the scalars  $\nu$  and  $\gamma$ . In addition to the values we previously suggested,  $\gamma = 0.99$  and  $\nu = 1$  are appropriate.

For Algorithm 14.4.3, we have the following analogy of Lemma 14.4.1.

**Lemma 14.4.4** Suppose that  $\mu_k$  converges to zero as  $k$  increases when Algorithm 14.4.3 is executed. Then the product  $\mu_k\|y_k\|$  converges to zero.

**Proof.** Let  $\mathcal{K} = \{k_0, k_1, k_2, \dots\}$  be the iterations on which

$$\|y_{k+1}^E\| \leq \nu(\mu_{k+1})^{-\gamma} + \theta_{k+1} \quad (14.4.29)$$

and consequently on which  $y_{k+1} = y_{k+1}^E$ . Then, from (14.4.29),

$$\mu_{k+1}\|y_{k+1}\| \leq \nu(\mu_{k+1})^{1-\gamma} + \mu_{k+1}\theta_{k+1}. \quad (14.4.30)$$

If  $\mathcal{K}$  is finite,  $y_k$  will be fixed for all  $k$  sufficiently large and the result is immediate. If  $\mathcal{K}$  is infinite, for any  $k_i < k \leq k_{i+1}$ ,  $y_k = y_{k_{i+1}}$  and  $\mu_k \leq \mu_{k_{i+1}}$ . Hence, from (14.4.30)

$$\mu_k\|y_k\| \leq \mu_{k_{i+1}}\|y_{k_{i+1}}\| \leq \nu(\mu_{k_{i+1}})^{1-\gamma} + \mu_{k_{i+1}}\theta_{k_{i+1}}. \quad (14.4.31)$$

By hypothesis, the right-hand side of (14.4.31) can be made arbitrarily small by choosing  $k_i$  large enough, and so  $\mu_k \|y_k\|$  converges to zero.  $\square$

We now show that a result analogous to Theorem 14.4.2 holds for Algorithm 14.4.3 under the same relatively weak conditions.

**Theorem 14.4.5** Assume that AW.1 holds. Let  $x_*$  be any limit point of the sequence  $\{x_k\}$  generated by Algorithm 14.4.3 for which AO.1c and AI.1 hold, and let  $\mathcal{K}$  be the set of indices of an infinite subsequence of the  $x_k$  whose limit is  $x_*$ . Then conclusions (i)–(iv) of Lemma 14.2.2 hold.

**Proof.** The proof is essentially identical to that of Theorem 14.4.2, excepting that the premises in parts (a) and (b) of the proof are now, respectively, that  $\mu_k$  does, and does not, converge to zero, and that Lemma 14.4.4 is used in place of Lemma 14.4.1 in part (a) to deduce that  $\mu_k \|y_k - y_*\|$  converges to zero.  $\square$

Note that Theorem 14.4.5 remains true regardless of the actual choices of  $\{\omega_k\}$  and  $\{\eta_k\}$ . In practice, however, a scheme similar to Algorithm 14.4.2 will be used.

It is not surprising that if we wish to show that the penalty parameter is bounded away from zero, we will ultimately need to restrict the multiplier estimates we can use. To this end, we obtain the following simple corollary of Lemma 14.2.2.

**Corollary 14.4.6** Suppose that the conditions of Lemma 14.2.2 (iii) hold and that  $y_{k+1}^E$  is any Lagrange multiplier estimate for which

$$\|y_{k+1}^E - y_*\| \leq \kappa_{13} \|x_k - x_*\| + \kappa_{14} \omega_k, \quad (14.4.32)$$

for some positive constants  $\kappa_{13}$  and  $\kappa_{14}$  and all  $k \in K$  sufficiently large. Then there are positive constants  $\mu_{\max}$ ,  $\kappa_3$ ,  $\kappa_4$ ,  $\kappa_5$ ,  $\kappa_6$ , and an integer value  $k_0$  so that if  $\mu_{k_0} \leq \mu_{\max}$  then (14.2.16),

$$\|y_{k+1}^E - y_*\| \leq \kappa_5 \omega_k + \kappa_6 \mu_k \|y_k - y_*\|, \quad (14.4.33)$$

and (14.2.18) hold for all  $k \geq k_0$ , ( $k \in \mathcal{K}$ ).

**Proof.** Inequality (14.4.33) follows immediately from (14.4.32) and (14.2.16).  $\square$

As before, we can now show that the penalty parameter in Algorithm 14.4.3 will usually be bounded away from zero.

**Theorem 14.4.7** Suppose that the iterates  $\{x_k\}$  generated by Algorithm 14.4.3 under AA.10, with the tolerance updating rule given in Algorithm 14.4.2, converge to the single limit point  $x_*$ . Furthermore, suppose that AO.3b, AW.1b, and AI.1 hold, that  $\alpha_\eta$  and  $\beta_\eta$  satisfy (14.4.18) and (14.4.19), and that (14.4.32) holds for all  $k$  sufficiently large. Then there is a constant  $\mu_* > 0$  such that  $\mu_k \geq \mu_*$  for all  $k$ .

**Proof.** The proof closely mirrors that of Theorem 14.4.3. Suppose, otherwise, that  $\mu_k$  tends to zero. Then, Step 3 of the algorithm must be executed infinitely often. We aim to obtain a contradiction to this statement by showing that Step 2 is always executed for  $k$  sufficiently large. We note that our assumptions are sufficient for the conclusions of Theorems 14.4.2 and 14.4.5 to hold.

Firstly, we show that the Lagrange multiplier estimates  $\{y_k\}$  converge to  $y_*$ . The result is clear if the multipliers updates are accepted infinitely often, as each time the update is performed,  $y_{k+1} = y_{k+1}^E$  and assumption (14.4.32) guarantees that  $y_{k+1}^E$  converges to  $y_*$ . Suppose that the update is not accepted infinitely often. Then for all  $k$  sufficiently large,  $\|y_{k+1}^E\| > \nu\mu_{k+1}^{-\gamma}$ , which implies that  $\|y_{k+1}^E\|$  diverges. But this contradicts assumption (14.4.32) and hence  $y_k$  converges to  $y_*$ . Therefore, as before,  $\mu_k\|y_k - y_*\|$  tends to zero as  $k$  increases.

The proof is now almost identical to that given for Theorem 14.4.3. The only extra requirement we need to ensure is that the multiplier update  $y_{k+1} = y_{k+1}^E$  eventually always takes place. To do this we pick  $k_3$  as before so that (14.4.22) and (14.4.23) hold, but now additionally so that

$$\mu_k^\gamma \leq \frac{\nu}{\|y_*\| + \omega\kappa_{12}}. \quad (14.4.34)$$

Inequality (14.4.33) then gives that

$$\begin{aligned} \|y_{k+1}^E\| &\leq \|y_*\| + \kappa_5\omega_k + \kappa_6\mu_k\|y_k - y_*\| \\ &\leq \|y_*\| + \kappa_5\omega\mu_k^{\alpha_\omega} + \kappa_6\mu_k\|y_k - y_*\| \quad (\text{from (14.4.20)}) \\ &\leq \|y_*\| + \omega(\kappa_5\mu_k^{\alpha_\omega} + \kappa_6\mu_k) \quad (\text{from (14.4.24)}) \\ &\leq \|y_*\| + \omega\kappa_{12}\mu_k^\alpha \quad (\text{from (14.4.18)}) \\ &\leq \|y_*\| + \omega\kappa_{12} \quad (\text{as } \mu_k < 1 \text{ and } \alpha > 0) \\ &\leq \nu\mu_{k+1}^{-\gamma} + \theta_{k+1}, \end{aligned}$$

for all  $k > k_5$ , the last inequality following from (14.4.34) and because  $\mu_{k+1} \leq \mu_k$  and  $\theta_{k+1} \geq 0$ . Hence, the multiplier update  $y_{k+1} = y_{k+1}^E$  in Algorithm 14.4.3 will always take place when  $k \geq k_0$ .

So long as we replace  $y_{k_0+j+2} = y_{k_0+j+1}^F$  by  $y_{k_0+j+2} = y_{k_0+j+2}^E$ , and any mention of (14.2.17) by (14.4.33), the remainder of the proof is identical to that of Theorem 14.4.3.  $\square$

There is one final, outstanding issue, namely, from where do we start the inner minimization (14.4.4)? If the penalty parameter converges to zero, the alternative “Newton” correction suggested at the end of Section 14.3 is appropriate. However, in view of Theorems 14.4.3 and 14.4.7, it is likely that the penalty parameter will be bounded away from zero. In this case, the obvious choice  $x_{k+1}^s = x_k$  turns out to be a reasonable choice.

To see why, we suppose that the penalty parameter has reached its least value  $\mu_*$ , and thus that Step 3 is no longer being executed. Then we have that both (14.4.5) and (14.4.6) hold for all subsequent iterations. Furthermore, the Step 2 update rules

$$\omega_{k+1} = \omega_k \mu_*^{\beta_\omega} \quad \text{and} \quad \eta_{k+1} = \eta_k \mu_*^{\beta_\eta}$$

will be in operation, and thus there are constants  $\kappa_{15}$  and  $\kappa_{16}$  such that

$$\omega_k = \kappa_{15} \mu_*^{\beta_\omega k} \quad \text{and} \quad \eta_k = \kappa_{16} \mu_*^{\beta_\eta k} \quad (14.4.35)$$

for all  $k$ . In view of conclusion (iv) of Lemma 14.2.2 and (14.4.35), it then follows that

$$\|x_k - x_*\| \leq \kappa_{17} \mu_*^{\min[\beta_\omega, \beta_\eta]k} \quad (14.4.36)$$

for some  $\kappa_{17}$ , and therefore the rate of convergence is asymptotically at least R-linear.

Now consider the gradient of the augmented Lagrangian function at the point  $x_{k+1}^s = x_k$  at the start of iteration  $k + 1$ . We have that

$$\nabla_x \Phi(x_{k+1}^s, y_{k+1}, \mu_*) = \nabla_x \Phi(x_k, y_k, \mu_*) + A^T(x_k)(y_{k+1} - y_k). \quad (14.4.37)$$

For Algorithm 14.4.1, we then have from (14.4.5)–(14.4.7), (14.4.35), and (14.4.37) that

$$\begin{aligned} \|\nabla_x \Phi(x_{k+1}^s, y_{k+1}, \mu_*)\| &\leq \|\nabla_x \Phi(x_k, y_k, \mu_*)\| + \frac{1}{\mu_*} \|A^T(x_k)c(x_k)\| \\ &\leq \kappa_{15} \mu_*^{\beta_\omega k} + 2\kappa_{16} \kappa_{\text{boa}} \mu_*^{\beta_\eta k - 1} \end{aligned} \quad (14.4.38)$$

for all  $k$  sufficiently large, where  $\kappa_{\text{boa}} = \|A(x_*)\|$ . Thus the gradient of the augmented Lagrangian at the starting point  $x_{k+1}^s$  converges to zero as  $k$  increases.

For Algorithm 14.4.3, so long as (14.4.32) is satisfied, (14.4.35) and (14.4.36) imply that

$$\|y_{k+1}^E - y_*\| \leq \kappa_{13} \kappa_{17} \mu_*^{\min[\beta_\omega, \beta_\eta]k} + \kappa_{14} \kappa_{15} \mu_*^{\beta_\omega k}. \quad (14.4.39)$$

As the algorithm eventually resorts only to Step 2, either  $y_{k+1} = y_k = y$  for some  $y$  and all sufficiently large  $k$ , or the multiplier  $y_{k+1}^E$  must be accepted for an infinite number of  $k$ . In the former case, we have from (14.4.5) and (14.4.35) that

$$\|\nabla_x \Phi(x_{k+1}^s, y, \mu_*)\| = \|\nabla_x \Phi(x_{k+1}^s, y_{k+1}, \mu_*)\| = \|\nabla_x \Phi(x_{k+1}^s, y_k, \mu_*)\| \leq \kappa_{15} \mu_*^{\beta_\omega k}, \quad (14.4.40)$$

and, on taking the limit as  $k$  tends to infinity, that the gradient of the augmented Lagrangian at the starting point  $x_{k+1}^s$  converges to zero and that  $y = y_*$ . In the latter

case, it follows from (14.4.39) that the limit  $y_*$  of  $y_{k+1}^E$  must satisfy

$$\|y_*\| \leq \nu \mu_*^{-\gamma}.$$

Thus, so long as  $\theta_k$  converges to zero more slowly than the right-hand side of (14.4.39),  $y_{k+1} = y_{k+1}^E$  for all sufficiently large  $k$ , and thus

$$\|y_{k+1} - y_k\| \leq (1 + \mu_*^{\min[\beta_\omega, \beta_\eta]}) \kappa_{13} \kappa_{17} \mu_*^{\min[\beta_\omega, \beta_\eta](k-1)} + (1 + \mu_*^{\beta_\omega}) \kappa_{14} \kappa_{15} \mu_*^{\beta_\omega(k-1)}.$$

Combining this with (14.4.5), (14.4.35), and (14.4.37) finally reveals that

$$\begin{aligned} \|\nabla_x \Phi(x_{k+1}^S, y_{k+1}, \mu_*)\| &\leq \kappa_{15} \mu_*^{\beta_\omega k} \\ &\quad + 2\kappa_{\text{boa}}(1 + \mu_*^{\min[\beta_\omega, \beta_\eta]}) \kappa_{13} \kappa_{17} \mu_*^{\min[\beta_\omega, \beta_\eta](k-1)} \\ &\quad + 2\kappa_{\text{boa}}(1 + \mu_*^{\beta_\omega}) \kappa_{14} \kappa_{15} \mu_*^{\beta_\omega(k-1)}, \end{aligned} \quad (14.4.41)$$

and, once again, the gradient of the augmented Lagrangian at the starting point  $x_{k+1}^S$  converges to zero as  $k$  increases.

Now suppose we take a Newton step  $s_{k+1}^S$  from  $x_{k+1}^S$ , and that

**AW.3**  $\nabla_{xx} \Phi(x_*, y_*, \mu_*)$  is positive definite.

Then, since  $x_{k+1}^S$  converges to  $x_*$ , and because of AW.3, we expect that

$$\|\nabla_x \Phi(x_{k+1}^S + s_{k+1}^S, y_{k+1}, \mu_*)\| \leq \kappa_{18} \|\nabla_x \Phi(x_{k+1}^S, y_{k+1}, \mu_*)\|^2 \quad (14.4.42)$$

for some  $\kappa_{18}$  and all  $k$  sufficiently large, using Theorems 3.3.1 and 3.3.2 (p. 52). Thus combining (14.4.38)/(14.4.40)/(14.4.41) and (14.4.42), we have that

$$\|\nabla_x \Phi(x_{k+1}^S + s_{k+1}^S, y_{k+1}, \mu_*)\| \leq \omega_{k+1} = \kappa_{15} \mu_*^{\beta_\omega(k+1)}$$

for all  $k$  sufficiently large so long as

$$\beta_\omega < 2\beta_\eta. \quad (14.4.43)$$

Hence, for sufficiently large  $k$ , a single Newton step suffices to satisfy the Step 1 stopping rule (14.4.5). Moreover, since the Hessian of the augmented Lagrangian function is positive definite in a neighbourhood of  $x_*$ , it is straightforward to show that the Newton step will be an acceptable trust-region step so long as the initial trust-region radius for each subproblem (14.4.4) is uniformly bounded away from zero.

This is important, as it shows that the at-worst asymptotic R-linear rate of convergence of the  $\{x_k\}$  is the true rate of convergence for the method. We note that (14.4.19) and (14.4.43) are consistent and that the values we suggested earlier satisfy both restrictions. Notice that it is only for this result that it is important to have a nonzero sequence  $\{\theta_k\}$  in Algorithm 14.4.3; all the other results of this section hold if  $\theta_k \equiv 0$  for all  $k$ .

## Notes and References for Section 14.4

The augmented Lagrangian method was proposed independently by Hestenes (1969) and Powell (1969), partly as a reaction to the perceived unfortunate side-effects associated with ill-conditioning of the simpler differentiable penalty and barrier functions (Lootsma, 1969; Murray, 1971b). Indeed, Powell showed, using a very simple device, how to ensure that the penalty parameter does not converge to zero and hence that the resulting ill-conditioning does not occur. A similar device is employed in the algorithms that we described in this section with the same consequence.

The most comprehensive references on augmented Lagrangians are the paper by Tapia (1977) and the book by Bertsekas (1982a). Globally convergent methods have been given by Powell (1969), Rockafellar (1976a), Bertsekas (1982a), Polak and Tits (1980), Yamashita (1982), Bartholomew-Biggs (1987), Hager (1987), and Conn, Gould, and Toint (1991b). Powell's method requires that the augmented Lagrangian be minimized exactly for fixed values of the multipliers and parameters. The multiplier estimates are guaranteed to be bounded but convergence is only established in the case where the underlying nonlinear program has a unique solution. Rockafellar, Bertsekas, and Polak and Tits allow inexact minimization of the augmented Lagrangian function but they require that the Lagrange multiplier estimates remain bounded—the methods differ in the stopping criteria used. Hager is slightly more restrictive in that he considers a particular multiplier update and specifies the method used for approximately minimizing the augmented Lagrangian function. His analysis also assumes that a subsequence of the Lagrange multipliers estimates converges. Yamashita and Bartholomew-Biggs are more specific in the method used for the inner minimization calculation—an appropriate quadratic program is solved—but their methods allow for more frequent updating of the penalty parameter and multiplier estimates. Yamashita establishes convergence under the assumption that the Lagrange multipliers for the quadratic programming problem stay bounded; the possibility of proving convergence for Bartholomew-Biggs's method under similar circumstances is only hinted at, although encouraging numerical results are presented. The methods analysed here are basically those proposed by Conn, Gould, and Toint, and examples are given there which show that Theorems 14.4.3 and 14.4.7 do not hold if the sequence  $\{x_k\}$  has more than one limit point. It is also easy to show that there is but a single limit point so long as AC1.d holds. As before, in view of Section 6.6, an appropriate trust-region method may be used to ensure that the terminating point  $x_k$  at the end of Step 1 of Algorithms 14.4.1 and 14.4.3 not only satisfies (14.4.5) but is such that  $\nabla_{xx}\Phi(x_k, y_k, \mu_k)$  is positive semidefinite. Since this condition is equivalent to  $K(x_k, y_k, \mu_k)$  having precisely  $m$  negative eigenvalues (see the notes at the end of Section 3.2.2), the same will be true of  $K(x_*, y_*, \mu_*)$ . However, this is not sufficient to guarantee that  $(x_*, y_*)$  will be a second-order critical point for (14.1.2) unless  $\mu_*$  is itself sufficiently small. This and (14.4.36), then, give two reasons for continuing to reduce  $\mu_k$  as much as possible in Step 2; of course, care must be taken because of the potential to introduce significant ill-conditioning.

Interest in augmented Lagrangians declined with the introduction of successive quadratic programming (SQP) techniques but has more recently regained some popularity; see, for example, the papers of Schittkowski (1981) and Gill, Murray, Saunders, and Wright (1992), which combine SQP with an augmented Lagrangian merit function. Both these methods are not *pure* augmented Lagrangian techniques since they perform a linesearch on the augmented Lagrangian as a function of both the position,  $x$ , and the multipliers,  $y$ , in contrast to the method described here.

Bertsekas (1982b) and others have remarked that augmented Lagrangian methods using first-order multiplier updates are particularly attractive for large problems. It is worth noting, however, that the Hessian matrix of the augmented Lagrangian function may be less sparse than for the Lagrangian function because of the last term in (14.4.2). While this is certainly true, we note that variables that appear *nonlinearly* in the constraint functions give rise to nonzeros in the same positions in the Hessians of both the Lagrangian and augmented Lagrangian function. Therefore, it may well be worth using such an approach but treating the linear constraint explicitly.

When simple bound constraints,  $x_l \leq x \leq x_u$ , are present as well as general equality constraints  $c(x) = 0$ , Conn, Gould, and Toint (1991b) propose that the test (14.4.5) in Algorithms 14.4.1 and 14.4.3 be replaced by the requirement that  $x_l \leq x_k \leq x_u$  and that

$$\|[x_k - P_C(x_k - \nabla_x \Phi(x_k, y_k, \mu_k))]\| \leq \omega_k, \quad (14.4.44)$$

where  $P_C$  is defined by (12.1.2) (p. 442). The intention is that an approximate solution of the subproblem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \Phi(x, y_k, \mu_k) \quad \text{subject to} \quad x_l \leq x \leq x_u$$

should be found in Step 1 of the modified algorithms, using one of the methods described in Section 12.2. All of the results from our current section apply in this more general setting, and, additionally, Conn, Gould, and Toint (1992c) show that the constraint identification properties described in Section 12.3 follow since (12.1.19) (p. 450) used in (14.4.44) is a suitable criticality measure. The resulting algorithm forms the basis of the large-scale nonlinear programming package **LANCELOT** (see, Conn, Gould, and Toint, 1992b).

This framework was further extended by Conn, Gould, Sartenaer, and Toint (1996a) to the case where the feasible set is defined by a combination of general equality and linear constraints, and for which the general constraints are handled by an augmented Lagrangian and the linear constraints are treated explicitly in the subproblem of Step 1. As before, each subproblem is considered solved when a suitable criticality measure—this time any of those considered in Section 12.1—is sufficiently small, and all of the results in the current section may be generalized. This framework also allows for a different penalty parameter for each constraint, which is advantageous for badly scaled problems.

Augmented Lagrangian methods for general problems of the form (14.1.1), based on the function (14.3.34), have been developed and analysed by Rockafellar (1974) and Bertsekas (1982a). First-order Lagrange multiplier estimates for the inequality constraints in this case are

$$[y_y^F(x, \mu)]_{\mathcal{I}} = y_{\mathcal{I}} - \min [y_{\mathcal{I}}, c_{\mathcal{I}}(x)/\mu],$$

while least-squares Lagrange multiplier estimates are now defined to be the minimum-norm solution of the problem

$$\underset{y, y_{\mathcal{I}} \geq 0}{\text{minimize}} \quad \|\nabla_x f(x) - A^T(x)y\|_2.$$

Augmented Lagrangian methods for infinite-dimensional problems have been considered by many authors. See, for example, Fortin and Glowinski (1982) and Le Tallec and Glowinski (1989). The above-mentioned method of Conn, Gould, and Toint (1991b) has been extended and generalized by Sachs and Sartenaer (2000), who apply it for the solution of infinite-dimensional optimal control problems. Their algorithm is very close in spirit to Algorithm 14.4.3, but it allows for a range of values of the constraint violation where either the penalty parameter

is reduced or the multipliers are updated (while Algorithm 14.4.3 allows both), as well as allowing perturbed first-order multiplier estimates. The purpose of these modifications is to cover the discretization error that occurs in the updates.

## 14.5 Nonsmooth Exact Penalty Functions

Another important class of penalty functions is the *nonsmooth* or, as they are sometimes known, nondifferentiable penalty functions typified by<sup>239</sup>

$$\phi(x, \sigma) = f(x) + \sigma \sum_{i \in \mathcal{E}} \|c_i(x)\| + \sigma \sum_{i \in \mathcal{I}} \|c_i(x)^-\| \equiv f(x) + \sigma \|c(x)^{\mathcal{T}-}\| \quad (14.5.1)$$

for some appropriate monotonic norm,  $\|\cdot\|$ . As before, we have defined

$$c(x) = \begin{pmatrix} c(x)_{\mathcal{E}} \\ c(x)_{\mathcal{I}} \end{pmatrix} \text{ and } c(x)^{\mathcal{T}-} = \begin{pmatrix} c(x)_{\mathcal{E}} \\ c(x)_{\mathcal{I}}^- \end{pmatrix},$$

where  $c^-$  denotes the componentwise minimum of a vector  $c$  and zero, that is,  $c^- = \min[c, 0]$  (see Section 11.4). Notice that, in this section, we are considering the general problem (14.1.1), since the difficulties in the proofs we are about to give are somewhat obscured for the simpler problem (14.1.2). The most commonly occurring examples are the  $\ell_1$  and  $\ell_\infty$  penalty functions for which  $\|\cdot\| = \|\cdot\|_1$  and  $\|\cdot\|_\infty$ , respectively. Since the function (14.5.1) is nonsmooth, but the nonsmoothness arises simply by composing nonsmooth, convex functions with smooth ones, the methods we discussed in Chapter 11 are appropriate, and we shall return to this in Section 15.3.2. Here, we merely aim to justify why such functions are penalty functions. Clearly the contribution from the constraints is more significant here when the constraints are almost satisfied than with smooth penalty functions, like those from Section 14.2 in which the contribution from the constraints is squared. In fact, this feature alone turns out to be most significant, for we have the following result.

**Theorem 14.5.1** Suppose that AW.1 holds, that  $x_*$  and  $y_*$  are such that  $x_*$  is feasible for (14.1.1), and that

$$\sigma \geq \|y_*\|_D, \quad (14.5.2)$$

where the subscript D indicates the vector dual norm (2.3.1) (p. 21). Then if  $x_*$  and  $y_*$  satisfy the second-order sufficiency conditions AO.3 for (14.1.1),  $x_*$  also satisfies the second-order sufficiency conditions AO.3n for a local minimizer of  $\phi(x, \sigma)$ . If, in addition,

$$\sigma > \|y_*\|_D, \quad (14.5.3)$$

the second-order sufficiency conditions for the two problems are equivalent.

---

<sup>239</sup>Notice that, for convenience, we are using the penalty parameter  $\sigma \stackrel{\text{def}}{=} 1/\mu$  instead of  $\mu$  here.

**Proof.** Compare the second-order sufficiency conditions for the two problems. The first-order criticality conditions (3.2.5)–(3.2.8) (pp. 40 and 41) for (3.2.4) are that

$$\nabla_x f(x_*) - A^T(x_*)y_* = 0, \quad c_{\mathcal{E}}(x_*) = 0, \quad \text{and} \quad c_{\mathcal{I}}(x_*) \geq 0, \quad (14.5.4)$$

where

$$y_* = \begin{pmatrix} y_{\mathcal{E}*} \\ y_{\mathcal{I}*} \end{pmatrix}, \quad y_{\mathcal{I}*} \geq 0, \quad \text{and} \quad \langle c(x_*), y_* \rangle = 0, \quad (14.5.5)$$

while those for the minimization of  $\phi(x, \sigma)$  are, using (3.1.6) (p. 34), that

$$\nabla_x f(x_*) + \sigma A^T(x_*)y^\Phi = 0, \quad (14.5.6)$$

where Corollary 11.4.2 (p. 430) implies that  $y^\Phi$  satisfies the conditions

$$y^\Phi = \begin{pmatrix} y_{\mathcal{E}}^\Phi \\ y_{\mathcal{I}}^\Phi \end{pmatrix}, \quad \langle c(x_*), y^\Phi \rangle = 0, \quad y_{\mathcal{I}}^\Phi \leq 0, \quad \text{and} \quad \|y^\Phi\|_{\mathbb{D}} \leq 1.$$

Furthermore

$$c_{\mathcal{E}}(x_*) = 0 \quad \text{and} \quad c_{\mathcal{I}}(x_*) \geq 0 \quad (14.5.7)$$

since  $x_*$  is feasible. Making the connection

$$y_* = -\sigma y^\Phi,$$

it follows immediately that (14.5.4) and (14.5.5) are equivalent to (14.5.6) and (14.5.7) when (14.5.2) holds.

In view of conditions (3.2.13) and (3.2.23) (p. 48), it remains to show that the sets

$$\mathcal{N}_+ = \left\{ d \left| \begin{array}{l} \langle d, \nabla_x c_i(x_*) \rangle = 0 \quad \text{for all } i \in \mathcal{E} \cup \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j > 0\}, \\ \langle d, \nabla_x c_i(x_*) \rangle \geq 0 \quad \text{for all } i \in \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j = 0\}, \text{ and} \\ \|d\| = 1 \end{array} \right. \right\}$$

in (3.2.12) (p. 41), where there is no loss of generality in normalizing  $s$  since  $s \neq 0$ , and

$$\mathcal{D} = \left\{ d \left| \max_{y \in \partial \|c(x_*)\|^{\mathcal{I}-}} \langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y \rangle = 0 \text{ and } \|d\| = 1 \right. \right\}$$

in (3.2.24) (p. 49), are identical when (14.5.3) holds, and that  $\mathcal{N}_+ \subset \mathcal{D}$  when (14.5.3) is weakened to (14.5.2).

Suppose  $d \in \mathcal{N}_+$  and that (14.5.2) holds. It follows immediately that  $\langle d, A^T(x_*)y_* \rangle = 0$ , and hence, from (14.5.4), that

$$\langle d, \nabla_x f(x_*) \rangle = 0. \quad (14.5.8)$$

Now consider any  $y$  in

$$\partial \|c(x_*)\|^{\mathcal{I}-} = \left\{ y = \begin{pmatrix} y_{\mathcal{E}} \\ y_{\mathcal{I}} \end{pmatrix} \left| \begin{array}{l} \langle y, c(x_*) \rangle = 0, \quad y_{\mathcal{I}} \leq 0, \quad \text{and} \\ \|y\|_{\mathbb{D}} \leq 1 \end{array} \right. \right\}.$$

As  $\langle y, c(x_*) \rangle = 0$  and  $y_{\mathcal{I}} \leq 0$ , it follows that  $y_i = 0$  for all  $i \notin \mathcal{A}(x_*)$ . Hence, since  $d \in \mathcal{N}_+$ , we have that

$$\langle d, A^T(x_*)y \rangle \leq 0 \quad (14.5.9)$$

for all  $y \in \partial\|c(x_*)^{\mathcal{I}-}\|$ . Thus recalling that  $y^\Phi = -y_*/\sigma$  lies in  $\partial\|c(x_*)^{\mathcal{I}-}\|$  and satisfies (14.5.6) when (14.5.2) holds, and combining (14.5.8) and (14.5.9), we have that

$$\langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y \rangle \leq \langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y^\Phi \rangle = 0$$

for all  $y \in \partial\|c(x_*)^{\mathcal{I}-}\|$ , and thus  $d \in \mathcal{D}$ .

Now suppose  $d \in \mathcal{D}$  and that (14.5.3) holds. As we have already seen,  $y^\Phi = -y_*/\sigma$  lies in  $\partial\|c(x_*)^{\mathcal{I}-}\|$  and satisfies

$$\langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y^\Phi \rangle = 0.$$

Thus

$$\begin{aligned} \langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y \rangle &\leq \langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y^\Phi \rangle \\ &\leq \max_{y \in \partial\|c(x_*)^{\mathcal{I}-}\|} \langle d, \nabla_x f(x_*) + \sigma A^T(x_*)y \rangle \\ &= 0, \end{aligned}$$

and therefore

$$\langle d, A^T(x_*)(y + y_*/\sigma) \rangle \leq 0, \quad (14.5.10)$$

for all  $y \in \partial\|c(x_*)^{\mathcal{I}-}\|$ . If  $i \in \mathcal{E}$  or if  $i \in \mathcal{A}(x_*) \cap \mathcal{I}$  and  $[y_*]_i > 0$ ,  $y = -y_*/\sigma \pm \epsilon e_i \in \partial\|c(x_*)^{\mathcal{I}-}\|$  for all sufficiently small  $\epsilon$  when (14.5.3) holds. Hence, in this case, (14.5.10) shows that

$$\langle d, \nabla_x c_i(x_*) \rangle = \langle d, A^T(x_*)e_i \rangle = 0.$$

On the other hand, if  $i \in \mathcal{A}(x_*) \cap \mathcal{I}$  and  $[y_*]_i = 0$ ,  $y = -y_*/\sigma - \epsilon e_i \in \partial\|c(x_*)^{\mathcal{I}-}\|$ , for all sufficiently small  $\epsilon$  when (14.5.3) holds, and thus (14.5.10) implies that

$$\langle d, \nabla_x c_i(x_*) \rangle = \langle d, A^T(x_*)e_i \rangle \geq 0.$$

Thus  $d \in \mathcal{N}_+$ . □

Theorem 14.5.1 implies that if we can find a feasible point that satisfies the second-order sufficiency conditions for the penalty function, (14.5.1), it will necessarily be a strict local solution for the related nonlinear programming problem (14.1.1) so long as (14.5.3) holds. Although we do not know methods that actually ensure second-order sufficiency conditions for either problem, this is at least an indication that, excepting pathological cases, the local minimizers of the two problems are related so long as the penalty parameter  $\sigma$  is large enough. But how important is condition (14.5.2)? As we shall now see, it is close to being essential.

**Theorem 14.5.2** Suppose that AW.1 holds and that  $x_*$  is a first-order critical point of the nonlinear program (14.1.1) with associated Lagrange multiplier  $y_*$ , but that

$$\sigma < \|y_*\|_{\mathbb{D}}. \quad (14.5.11)$$

Then  $x_*$  is not a local minimizer of  $\phi(x, \sigma)$ .

**Proof.** It follows immediately from (14.5.5) and (14.5.11) that  $y^* = -y_*/\sigma \notin \partial\|c(x_*)^{\mathcal{I}^-}\|$ , and thus that

$$\nabla_x f(x_*) + \sigma A^T(x_*) y^* = \nabla_x f(x_*) - A^T(x_*) y_* = 0$$

is not in  $\partial\phi(x_*, \sigma)$ . Thus Corollary 3.2.13 (p. 48) implies that  $x_*$  is not a first-order critical point of  $\phi(x, \sigma)$ .  $\square$

Thus we see the clear dependence of this class of penalty functions on the value of  $\sigma$ . For  $\sigma$  larger than  $\|y_*\|_{\mathbb{D}}$ , locally minimizing the penalty function is highly likely to produce a critical point for the underlying nonlinear program, while it is unlikely to do so if  $\sigma$  is smaller than this value. The fact that the penalty function is likely to provide critical points for the underlying problem for all sufficiently large  $\sigma$  is crucial, since it implies that the penalty parameter need not approach infinity to ensure convergence. Such a function is known as an *exact* penalty function, since a single minimization with a sufficiently large penalty parameter normally suffices to determine a critical point for the nonlinear program. This contrasts with the methods of Section 14.3, where parameters have to approach unfortunate limits, or those in Section 14.4, where the functions are exact only if the Lagrange multipliers take on their (unknown) optimal values. Notice also that the critical value of  $\sigma$  depends only on first-order information; by contrast, the critical value for the augmented Lagrangian function depends on the curvature of the functions involved. We illustrate the typical dependence of the exact penalty function upon the penalty parameter in Figure 14.5.1.

Barring pathological cases, so long as the penalty parameter is large enough, the only time that local minimizers of  $\phi$  do not yield critical points for the nonlinear program is when the former are infeasible for the latter; this will certainly happen when the underlying problem is infeasible, in which case the penalty function provides a “least infeasible” solution. That undesirable local minimizers do occur is illustrated by the simple example of finding a feasible point for the constraint  $1 + x^2 - 2x^4 = 0$  by minimizing  $|1 + x^2 - 2x^4|$ , which leads to three infeasible, isolated critical values  $x = 0, \pm\frac{1}{2}$ , but fortunately such examples appear to occur rarely in practice. Notice that this phenomenon may occur whenever AO.1c is violated, and as we saw on p. 578, it is not restricted to nonsmooth penalty functions.

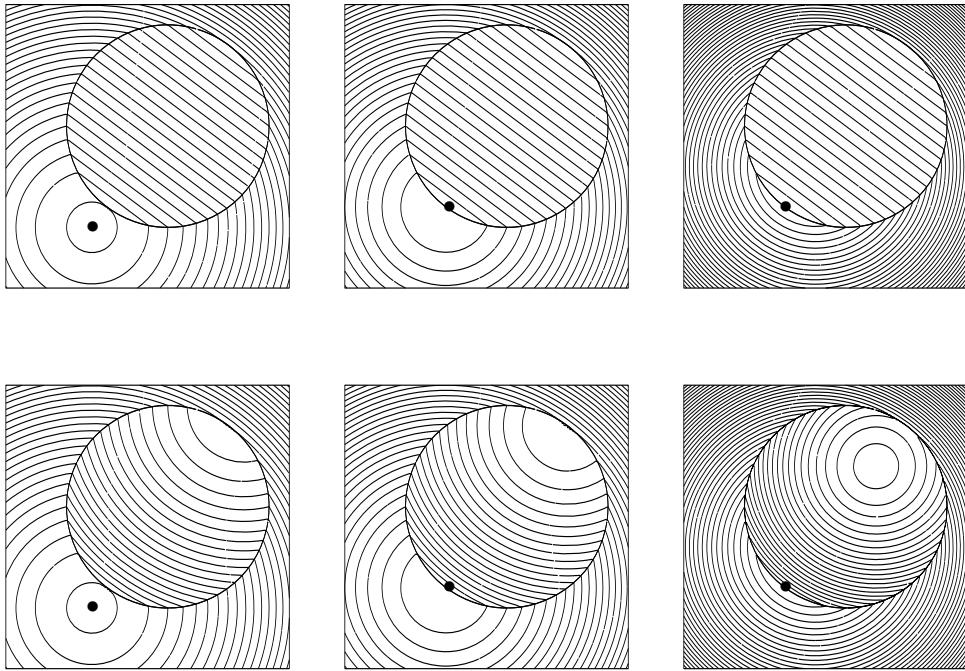


Figure 14.5.1: The top three figures illustrate the contours of the exact penalty function (14.5.1) corresponding to the two-dimensional problem: minimize  $f(x_1, x_2) = 3x_1 + 4x_2$  subject to the inequality constraint  $x_1^2 + x_2^2 \leq 1$ , for values  $\sigma = 2, 2.5 (= y_*)$ , and 5, respectively. The minimizer of the penalty function is indicated by a bullet ( $\bullet$ ), while the dark curve indicates the boundary of the feasible region. Notice how the minimizer of the penalty function does not correspond to that of the original problem in the leftmost figure and that the contours in the rightmost figure for infeasible points are significantly steeper than those in the middle figure. The bottom three figures illustrate the exact penalty function, with the same values of the penalty parameter, when the inequality constraint is replaced by the equality constraint  $x_1^2 + x_2^2 = 1$ .

Although Theorem 14.5.1 provides a realistic lower bound on the required value of the penalty parameter, the bound (14.5.3) depends, unfortunately, on a posteriori information concerning the optimal values of the Lagrange multipliers. A practical algorithm is unlikely to know suitable values and consequently must be capable of adjusting the parameter to correct for an unfortunate initial choice. Clearly, too small a value will likely produce a critical point for the penalty function that is not a critical point for the nonlinear program. It is also apparent, from Figure 14.5.1, that too large a value leads to an ill-conditioned penalty function that may prove difficult to minimize. Thus it is crucial that an automatic adjustment scheme does not increase the parameter far above the critical value  $\|y_*\|_{\mathbb{D}}$ .

A number of different methods have been suggested. In the first, if estimates  $y$  of the optimal Lagrange multipliers are computed during the minimization process, these may be used to decide whether to increase the penalty parameter. Specifically, if the current  $\sigma$  is smaller than the norm of such estimates, it is replaced by  $\max[\sigma + \epsilon_1, \|y\|_{\mathcal{D}} + \epsilon_2]$  for some appropriate positive constants  $\epsilon_1$  and  $\epsilon_2$ . Typical values might be  $\epsilon_1 = \epsilon_2 = 1$ .

In the second class of methods, rather than choosing  $\sigma$  on the basis of estimates of  $\|y^*\|_{\mathcal{D}}$ , the penalty parameter is chosen to ensure that the penalty function initially decreases in a given direction  $d$ . That is, we wish to ensure that  $\phi'_d(x, \sigma) < 0$  or, equivalently, that

$$\langle \nabla_x f(x), d \rangle + \sigma \|c(x)^{\mathcal{I}^-}\|'_d < 0.$$

This alternative is particularly appropriate when the chosen direction  $d$  has not been chosen directly from a model of the penalty function, but has been imposed for some other reason (see Section 15.3).

Thirdly, and probably most simply, we may choose to assess the suitability of a given  $\sigma_k$  only after an approximate first-order critical point,  $x_k$ , of  $\Phi(x, \sigma_k)$  has been determined. If the point is also feasible for the constraints, it is an approximate first-order critical point for the underlying nonlinear program (14.1.1). If not,  $\sigma_{k+1}$  should be increased by a significant amount—for instance, one might choose  $\sigma_{k+1} \geq \max[\tau\sigma_k, \sigma_k + \epsilon]$ , where  $\tau \geq 1$ ,  $\epsilon \geq 0$ , and  $\tau\epsilon > 0$ —and the minimization repeated. So long as  $\sigma_k$  converges to infinity if an infinite sequence proves necessary, and we choose our termination test carefully, then, at worst, we can be sure that we will converge to a first-order critical point for the infeasibility  $\|c(x)^{\mathcal{I}^-}\|$ . For example, suppose that  $\{\omega_k\}$  and  $\{\eta_k\}$  are two sequences of nonnegative values, whose limits are both zero. Now let  $\mu_k = 1/\sigma_k$ , pick  $x_k$  to ensure that

$$\|\mu_k \nabla_x f(x_k) + A^T(x_k) y_k\| \leq \omega_k, \quad (14.5.12)$$

where  $y_k \in \partial\|c(x_k)^{\mathcal{I}^-}\|$ , and set  $\sigma_{k+1} = \sigma_k$  if  $\|c(x_k)^{\mathcal{I}^-}\| \leq \eta_k$  but  $\sigma_{k+1} > \sigma_k$  if  $\|c(x_k)^{\mathcal{I}^-}\| > \eta_k$ . Then, so long as  $\{x_k\}$  is bounded and  $\sigma_k$  grows without bound if  $\limsup_{k \rightarrow \infty} \|c(x_k)^{\mathcal{I}^-}\| > 0$ , any infeasible limit point  $x_*$  will be first-order critical for  $\|c(x)^{\mathcal{I}^-}\|$ . This follows since, if  $\{x_k\}$ ,  $k \in \mathcal{K}$ , converges to the infeasible  $x_*$ , there will be a subsequence  $\mathcal{K}_y \subset \mathcal{K}$  for which  $\{y_k\}$ ,  $k \in \mathcal{K}_y$ , converges to  $y_* \in \partial\|c(x_*)^{\mathcal{I}^-}\|$ , because  $\{y_k\}$  is bounded. Additionally (14.5.12) implies that

$$A^T(x_*) y_* = 0, \quad (14.5.13)$$

which shows that  $x_*$  is indeed first-order critical for  $\|c(x)^{\mathcal{I}^-}\|$ . We must be cautious here that  $\eta_k$  is sufficiently large to allow  $\|c(x_k)^{\mathcal{I}^-}\|$  to converge to zero without increasing  $\sigma_k$  unnecessarily. Notice that if  $\mathcal{A}(x_*)$  denotes the set of constraints that are not strictly feasible at  $x_*$  (i.e., those in  $\mathcal{E}$  as well as those in  $\mathcal{I}$  for which  $c_i(x_*) \leq 0$ ), and if  $A_{\mathcal{A}(x_*)}(x_*)$  is of full rank, then  $x_*$  must be feasible. Suppose otherwise: then Corollary 11.4.2 (p. 430) and (14.5.13) together imply that  $[y_*]_{\mathcal{A}(x_*)} \neq 0$  and that  $A_{\mathcal{A}(x_*)}^T(x_*) [y_*]_{\mathcal{A}(x_*)} = 0$ , which is impossible if  $A_{\mathcal{A}(x_*)}(x_*)$  is of full rank.

Finally, if the constraint infeasibility appears to be decreasing significantly, it may well be that we should decrease the penalty parameter. The following automatic method is possible. Let  $\phi_0$  and  $\epsilon_\phi$  be any positive scalars and define the sequence  $\{\phi_k\}$  of decreasing values by

$$\phi_k = \min[\phi_{k-1}, \|c(x_k)^{\mathcal{I}^-}\|].$$

Then we may reset  $\sigma_k$  to any positive value we desire if  $\phi_k \leq \phi_{k-1} - \epsilon$ . Since the elements of  $\{\phi_k\}$  are bounded away from zero, a decrease in adjacent elements of the sequence of this type can only possibly occur a finite number of times. Notice that we need not decrease the penalty parameter—leaving the value unchanged or even increasing it is also permissible—but since the opportunity may not arise very often, it is at least worth contemplating if the parameter is too large.

## Notes and References for Section 14.5

The equivalence between the optimality conditions for nonconvex nonlinear programming problems and related penalty functions was first reported by Pietrzykowski (1969), and the results subsequently strengthened by Charalambous (1978), Han and Mangasarian (1979), Coleman and Conn (1980), Bazaraa and Goode (1982), and Huang and Ng (1994). Our development follows that of Fletcher (1987a, Chapters 12 and 14). Algorithms based directly on nonsmooth penalty functions include those of Conn (1973), Conn and Pietrzykowski (1977), Coleman and Conn (1982a, 1982b), Mine, Fukushima, and Tanaka (1984), and Fletcher (1987a), while such functions are frequently used as merit functions for SQP methods. We shall discuss these issues in more detail in Chapter 15.

Methods for automatically increasing the penalty parameter are considered by Powell (1978), Mayne and Maratos (1979), Sahba (1987), Pantoja and Mayne (1991), Yuan (1995), Burke (1992), and Mongeau and Sartenaer (1995). The procedure we outlined for decreasing the parameter is that of Sahba (1987).

Nonsmooth penalty functions have also been suggested when there are an infinite number of constraints. See Tanaka, Fukushima, and Hasegawa (1987), Tanaka, Fukushima, and Ibaraki (1988), and Tanaka (1999) for details.

## 14.6 Smooth Exact Penalty Functions

Although there are powerful methods for minimizing nonsmooth penalty functions, it would clearly be advantageous to build smooth, exact penalty functions, since there are many more methods for minimizing smooth functions. We shall once again focus on the equality problem (14.1.2). The obvious first attempt is to try to mimic the approach taken in Section 14.5 by considering a penalty function of the form

$$\Phi(x, \sigma) = f(x) + \sigma w(c(x)), \quad (14.6.1)$$

where  $w(c)$  is a nonnegative function for which  $w(0) = 0$ , but now to allow  $w$  to be differentiable. Unfortunately, it is clear that all such functions must satisfy  $\nabla_c w(0) = 0$ , as 0 is a minimizer of  $w$ , and thus that

$$\nabla_x \Phi(x_*, \sigma) = \nabla_x f(x_*) + \sigma(\nabla_x c(x_*))^T \nabla_c w(c(x_*)) = \nabla_x f(x_*)$$

for any critical point  $x_*$  of (14.1.2). Thus a critical point for the problem will not be a critical point for the penalty function except in the very unlikely case that  $\nabla_x f(x_*) = 0$ . Hence, such an approach is fatally flawed. In this section, we shall show that, despite this seeming setback, it is nonetheless still possible to construct smooth exact penalty functions.

We have already encountered most of the basic ingredients in Section 14.2, and it merely remains for us to combine them in an appropriate way. The central idea is to consider the augmented Lagrangian function (14.2.2), but to choose the Lagrange multipliers  $y$  to be the least-squares estimates (14.2.8) at every point  $x$ . Thus we consider the penalty function

$$\Phi(x, \sigma) = f(x) - \langle c(x), y(x) \rangle + \frac{1}{2}\sigma\|c(x)\|_2^2, \quad (14.6.2)$$

where the Lagrange multiplier estimates

$$y(x) = (A(x)A^T(x))^{-1}A(x)\nabla_x f(x)$$

are themselves functions of  $x$ , and we have written  $\sigma = 1/\mu$ . Unlike the methods we considered in Section 14.4, the intention now is that the function (14.6.2) should be minimized for a single value of  $\sigma$ , rather than for a sequence of  $\sigma$ 's and  $y$ 's. Notice that a fundamental requirement is that  $A(x)$  be of full rank at every point  $x$  we encounter, and this is a considerably stronger assumption than AO.1c. We now show that, under such an assumption, (14.6.2) is an exact penalty function.

**Theorem 14.6.1** Suppose that AW.1 holds, that  $x_*$  is feasible for (14.1.2), and that AO.1c holds there. Then there is a critical value,  $\sigma_{\min} \geq 0$ , such that the second-order sufficiency conditions for (14.1.2) and those for the problem of minimizing (14.6.2) are equivalent for all  $\sigma > \sigma_{\min}$ .

**Proof.** The gradient and Hessian of (14.6.2) are

$$\nabla_x \Phi(x, \sigma) = \nabla_x \ell(x, y(x, \sigma)) - (\nabla_x y(x))^T c(x) \quad (14.6.3)$$

and

$$\nabla_{xx} \Phi(x, \sigma) = \nabla_{xx} \ell(x, y(x, \sigma)) + \sigma A^T(x)A(x) - (\nabla_x y(x))^T A(x) - A^T(x)(\nabla_x y(x)), \quad (14.6.4)$$

where  $y(x, \sigma) = y(x) - \sigma c(x)$ . Since  $x_*$  is feasible,

$$c(x_*) = 0 \text{ and } y(x_*, \sigma) = y(x_*). \quad (14.6.5)$$

Thus (14.6.3) and (14.6.5) show that the first-order optimality conditions

$$0 = \nabla_x \Phi(x_*, \sigma) = \nabla_x \ell(x_*, y(x_*, \sigma)) - (\nabla_x y(x_*))^T c(x_*) = \nabla_x \ell(x_*, y(x_*)) \quad (14.6.6)$$

for a minimizer of (14.6.2) and the same conditions for (14.1.2) are identical provided we define  $y_* \stackrel{\text{def}}{=} y(x_*)$ . Furthermore, Lemma 14.2.1 and (14.6.6) show that

$$\nabla_x y(x_*) = (A^+(x_*))^T \nabla_{xx} \ell(x_*, y_*). \quad (14.6.7)$$

Thus, combining (14.6.4), (14.6.5), and (14.6.7), it remains for us to consider the Hessian

$$\nabla_{xx} \Phi(x_*, \sigma) = Q_*^T \nabla_{xx} \ell(x_*, y_*) Q_* - P_*^T \nabla_{xx} \ell(x_*, y_*) P_* + \sigma A^T(x_*) A(x_*), \quad (14.6.8)$$

where we have defined the orthogonal projection matrices

$$P_* = A^T(x_*) \left( A(x_*) A(x_*)^T \right)^{-1} A(x_*) \text{ and } Q_* = I - P_*.$$

Suppose that second-order sufficiency conditions for a minimizer of (14.6.2) are satisfied at  $x_*$ , that is, that  $\nabla_{xx} \Phi(x_*, \sigma)$  is positive definite. Then  $\langle s, \nabla_{xx} \Phi(x_*, \sigma) s \rangle > 0$  for all  $s \neq 0$ , in particular for all  $s$  satisfying  $A(x_*) s = 0$ . But then, since  $P_* s = 0$  and  $Q_* s = s$ , it follows from (14.6.8) that for all such  $s$ ,

$$\begin{aligned} 0 &< \langle s, \nabla_{xx} \Phi(x_*, \sigma) s \rangle \\ &= \langle s, Q_*^T \nabla_{xx} \ell(x_*, y_*) Q_* s \rangle - \langle s, P_*^T \nabla_{xx} \ell(x_*, y_*) P_* s \rangle + \sigma \langle s, A^T(x_*) A(x_*) s \rangle \\ &= \langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle \end{aligned}$$

and thus that  $x_*$  and  $y_*$  satisfy second-order sufficiency conditions for (14.1.2).

Conversely, any vector  $s$  may be expressed as  $s = s_n + A^T(x_*) s_a$ , where  $A(x_*) s_n = 0$ . For such a vector,  $P_* s = A^T(x_*) s_a$ ,  $Q_* s = s_n$ , and

$$\begin{aligned} \langle s, \nabla_{xx} \Phi(x_*, \sigma) s \rangle &= \langle s_n, \nabla_{xx} \ell(x_*, y_*) s_n \rangle - \langle s_a, A(x_*) \nabla_{xx} \ell(x_*, y_*) A^T(x_*) s_a \rangle \\ &\quad + \sigma \langle s_a, A(x_*) A^T(x_*) A(x_*) A^T(x_*) s_a \rangle. \end{aligned} \quad (14.6.9)$$

The second-order sufficiency conditions for (14.1.2) imply that

$$\langle s_n, \nabla_{xx} \ell(x_*, y_*) s_n \rangle \geq \kappa_{\text{sos}} \|s_n\|_2^2$$

for some  $\kappa_{\text{sos}} > 0$ . In addition, let

$$\begin{aligned} \kappa_{\text{aha}} &= \lambda_{\max}[A(x_*) \nabla_{xx} \ell(x_*, y_*) A^T(x_*)], \\ \text{and } \kappa_{\text{aat}} &= \lambda_{\min}[A(x_*) A^T(x_*)]^2 > 0. \end{aligned} \quad (14.6.10)$$

Then, combining (14.6.9)–(14.6.10), we find that

$$\langle s, \nabla_{xx} \Phi(x_*, \sigma) s \rangle \geq \kappa_{\text{sos}} \|s_n\|_2^2 + (\sigma \kappa_{\text{aat}} - \kappa_{\text{aha}}) \|s_a\|_2^2. \quad (14.6.11)$$

If we now choose any

$$\sigma > \sigma_{\min} \stackrel{\text{def}}{=} \kappa_{\text{aha}} / \kappa_{\text{aat}},$$

inequality (14.6.11) implies that

$$\langle s, \nabla_{xx} \Phi(x_*, \sigma) s \rangle > 0$$

provided that  $s \neq 0$ . Thus the second-order sufficiency conditions for a minimizer of (14.6.2) are satisfied at  $x_*$  for all  $\sigma > \sigma_{\min}$ .  $\square$

Notice that, unlike the nonsmooth exact penalty functions considered in the previous section, the critical value of the penalty parameter is dependent on second-order rather than first-order information. We illustrate the penalty function in Figure 14.6.1.

As the reader may have already noticed, the principal, practical disadvantage of the smooth exact penalty function (14.6.2) is that its values depend on derivative information. In particular, to calculate the function at a given  $x$ , it is necessary to solve a least-squares problem in order to evaluate  $y(x)$ . If a derivative-based method is

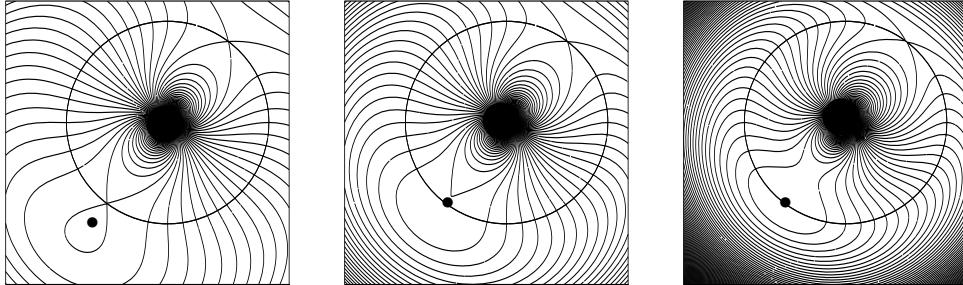


Figure 14.6.1: The figures illustrate the contours of the exact penalty function (14.6.2) corresponding to the two-dimensional problem: minimize  $f(x_1, x_2) = 3x_1 + 4x_2$  subject to the equality constraint  $x_1^2 + x_2^2 = 1$  for values  $\sigma = 0.5, 1.5$ , and  $5$ , respectively. The critical value of  $\sigma$  is  $\sigma_{\min} = 1.0$  for this problem. The minimizer of the penalty function is indicated by a bullet ( $\bullet$ ), while the dark curve indicates the boundary of the feasible region. Note the singularity at the origin, at which point the constraint Jacobian is rank deficient. Notice also how the minimizer of the penalty function does not correspond to that of the original problem in the leftmost figure—the penalty function merely has a saddle point at the solution to the latter—while, as before, the contours in the rightmost figure for infeasible points are significantly steeper than those in the middle figure.

used to minimize the function, equations (14.6.3) and (14.6.4) show that the gradient and Hessian of (14.6.2) depend on second and third derivatives of the problem functions, respectively. It is possible to replace the exact second derivatives of the penalty function with a good approximation by disregarding the last term in (14.6.4), which vanishes as we approach feasibility. The major contemporary use for (14.6.2) is as a merit function for other methods, and we shall consider this further in Chapter 15.

## Notes and References for Section 14.6

The smooth exact penalty function is due to Fletcher (1970a). Fletcher (1973) prefers the variant

$$\Phi(x, \sigma) = f(x) - \langle c(x), y(x) \rangle + \sigma \left\langle c(x), \left( A(x)A^T(x) \right)^{-1} c(x) \right\rangle, \quad (14.6.12)$$

which really is (14.6.2) with a different metric for the norm in the last term. Fletcher shows that this can be rewritten as

$$\Phi(x, \sigma) = f(x) - \langle c(x), v(x) \rangle,$$

where

$$v(x) = \arg \min_v \frac{1}{2} \|A^T(x)v - \nabla_x f(x)\|_2^2 + \sigma \langle v, c(x) \rangle. \quad (14.6.13)$$

Further generalizations of (14.6.2) are possible, and most are covered by the function

$$f(x) - \langle c(x), u(x) \rangle + \sigma \frac{\|c(x)\|_2^2}{a(x)}, \quad (14.6.14)$$

where

$$(A(x)A^T(x) + \gamma I)u(x) = A(x)\nabla_x f(x)$$

and  $0 < a(x) \leq \alpha$  for some  $\alpha > 0$  and  $\gamma \geq 0$ . Such developments are originally due to Di Pillo and Grippo (1985) and have the significant advantage that the penalty function is well defined when  $\gamma > 0$  even if  $A(x)$  is not of full rank.

It is relatively straightforward to adapt the functions (14.6.2) or (14.6.12) to handle inequality constraints. The most obvious generalization is to replace (14.2.8) by

$$y(x) = \arg \min_{y \geq 0} \|A^T(x)y - \nabla_x f(x)\|_2. \quad (14.6.15)$$

Fletcher (1973) gives a simple generalization of the function (14.6.12) in the inequality case by replacing (14.6.13) with

$$v(x) = \arg \min_{v \geq 0} \frac{1}{2} \|A^T(x)v - \nabla_x f(x)\|_2^2 + \sigma \langle v, c(x) \rangle. \quad (14.6.16)$$

Since the Lagrange multipliers (14.6.15) or (14.6.16) may not be differentiable, Glad and Polak (1979) prefer to use the function (14.3.34) together with the differentiable multiplier updates

$$y(x) = \arg \min_y \frac{1}{2} \|A^T(x)y - \nabla_x f(x)\|_2^2 + \langle y, c(x) \rangle^2.$$

Boggs, Tolle, and Kearsley (1991) choose to introduce extra slack variables  $s$  in order to replace the inequality constraints with the equations  $c_i(x) - \frac{1}{4}s_i^2 = 0$ . Introducing these variables into (14.6.12) and setting  $z_i = \frac{1}{4}s_i^2 \geq 0$ , they propose the penalty function

$$f(x) - \langle c(x) - z, y(x) \rangle + \sigma \left\langle c(x) - z, \left( A(x)A^T(x) + Z \right)^{-1} (c(x) - z) \right\rangle,$$

where  $Z$  is the diagonal matrix with entries  $z_i$  and

$$(A(x)A^T(x) + Z)y(x) = A(x)\nabla_x f(x).$$

It is straightforward to design updates for  $z$  which ensure that  $z \geq 0$ . A version of (14.6.14) appropriate for inequality constraints is given by Di Pillo and Grippo (1986), Lucidi (1992), and Di Pillo, Facchinei, and Grippo (1992). See also Han and Mangasarian (1983).

Methods for automatically adjusting the penalty parameter are given by Mukai and Polak (1975) for the equality-constrained case and Glad and Polak (1979) for the general case.

# Chapter 15

---

## Sequential Quadratic Programming Methods

---

### 15.1 Introduction

At last, we turn to probably the most powerful, highly regarded methods for solving smooth nonconvex, nonlinear optimization problems involving nonlinear constraints, sequential quadratic programming (SQP)—or, as they are sometimes called, successive or recursive quadratic programming (RQP)—methods. As the names imply, such methods solve a sequence of quadratic programs. But, which quadratic programs? And why? The key to this is, as always, Newton’s method. The goal is to produce rapidly convergent methods for the problem at hand. But in all of our haste, it is important not to forget to consider how best to embed these methods within a globally convergent framework. It is this issue that is particularly delicate.

Rather than sneaking up on the solution in a rather indirect way, as do barrier methods (see Chapter 13) or differentiable penalty methods (see Chapter 14), SQP methods aim directly for a first-order critical point of the given nonlinear program

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \quad \text{for } i \in \mathcal{E} \\ & && \text{and} \quad c_i(x) \geq 0 \quad \text{for } i \in \mathcal{I}. \end{aligned} \tag{15.1.1}$$

Here, as before,  $\mathcal{E}$  and  $\mathcal{I}$  are, respectively, disjoint sets of the indices of equality and inequality constraints, while  $f$  and the  $c_i$  smoothly map  $\mathbb{R}^n$  into  $\mathbb{R}$ .

We start our exposition by, once again, considering the simplified problem

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && c_i(x) = 0 \quad \text{for } i \in \mathcal{E}, \end{aligned} \tag{15.1.2}$$

which only involves equality constraints, as it is here that the quadratic programming connection is most pronounced. Only then can we make the slightly tenuous connection with the general problem (15.1.1).

## 15.2 What Is Sequential Quadratic Programming?

### 15.2.1 Methods for Problems with Equality Constraints

Consider the equality-constrained problem (15.1.2). Let  $c(x)$  be the vector whose components are the  $c_i(x)$  for  $i \in \mathcal{E}$ , and denote the Jacobian of  $c(x)$  by  $A(x)$ . Furthermore, write  $g(x)$  for the gradient of  $f(x)$ , define the Lagrangian function

$$\ell(x, y) = f(x) - \sum_{i \in \mathcal{E}} y_i c_i(x),$$

and let

$$K(x, y) = \begin{pmatrix} H(x, y) & A^T(x) \\ A(x) & 0 \end{pmatrix}, \quad (15.2.1)$$

where  $y$  is any set of auxiliary variables and  $H(x, y) \stackrel{\text{def}}{=} \nabla_{xx}\ell(x, y)$ . We recall from Section 3.2.2 that the first-order optimality conditions for (15.1.2) are that there is an  $x$  and a vector of Lagrange multipliers  $y$  for which

$$\nabla_x \ell(x, y) = 0 \quad \text{and} \quad c(x) = 0. \quad (15.2.2)$$

Suppose that  $(x_k, y_k)$  are estimates of the critical values  $(x, y)$ . Then the Newton equations for suitable corrections  $(s_k, s_k^y)$  to these estimates are simply<sup>240</sup>

$$\begin{pmatrix} H_k & A^T(x_k) \\ A(x_k) & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -s_k^y \end{pmatrix} = - \begin{pmatrix} \nabla_x \ell(x_k, y_k) \\ c(x_k) \end{pmatrix}, \quad (15.2.3)$$

where  $H_k = H(x_k, y_k)$ . But, in view of the equivalence between (4.4.8) (p. 71) and (4.4.14) (p. 72), the solution  $s_k$  to (15.2.3) is a critical point of the quadratic programming problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H_k s \rangle + \langle \nabla_x \ell(x_k, y_k), s \rangle \quad (15.2.4a)$$

$$\text{subject to} \quad A(x_k)s + c(x_k) = 0, \quad (15.2.4b)$$

while  $-s_k^y$  are its Lagrange multipliers. Perhaps, more revealingly, we may rewrite (15.2.3) as

$$\begin{pmatrix} H_k & A^T(x_k) \\ A(x_k) & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -y_{k+1} \end{pmatrix} = - \begin{pmatrix} g(x_k) \\ c(x_k) \end{pmatrix},$$

where  $y_{k+1} = y_k + s_k^y$ , and thus  $s_k$  is equivalently a critical point of the alternative quadratic programming problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H_k s \rangle + \langle g(x_k), s \rangle \quad (15.2.5a)$$

$$\text{subject to} \quad A(x_k)s + c(x_k) = 0, \quad (15.2.5b)$$

---

<sup>240</sup>The minus sign in front of the solution component  $s_k^y$  is simply so that we can write the Newton equations with a symmetric coefficient matrix.

with  $-y_{k+1}$  being its Lagrange multipliers. In view of this, another interpretation of (15.2.5) which is sometimes made is that the constraints are a linearization, or first-order Taylor approximation, of  $c(x_k + s) = 0$  about  $x_k$ , while (15.2.5a) is roughly a second-order Taylor approximation of the objective in which curvature from the constraints, which was ignored in the linearization of  $c(x_k + s) = 0$ , is instead included in the Hessian of the model. We do not find such a justification totally convincing and prefer instead to view the equivalence between (15.2.3) and (15.2.5) merely as a convenient tool.

The main reason for the considerable interest in such methods is, of course, their potential for fast convergence. Formally, we have the following result.

**Theorem 15.2.1** Suppose that AW.1b holds in some neighbourhood  $\Omega$  of the first-order critical point  $(x_*, y_*)$  of (15.1.2) and that the second-order nonsingularity assumption AO.3b holds there. Then,

- (i) there is a neighbourhood  $\mathcal{X} \subset \Omega$  of  $x_*$  for which the sequence  $\{x_k\}$ , generated by the iteration  $x_{k+1} = x_k + s_k$  using the correction  $s_k$  given by (15.2.5) and any  $\{y_k\}$  converging to  $y_*$ , converges Q-superlinearly to  $x_*$  from any starting point  $x_0$  in  $\mathcal{X}$ . If  $\|y_k - y_*\| = O(\|x_k - x_*\|)$ , the convergence is quadratic; and
- (ii) there is a neighbourhood  $\mathcal{X} \subset \Omega$  of  $x_*$  and another neighbourhood  $\mathcal{Y}$  of  $y_*$  for which the sequence  $\{(x_k, y_k)\}$ , generated by the iteration  $x_{k+1} = x_k + s_k$  using the correction  $s_k$  given by (15.2.5) together with its Lagrange multipliers  $y_{k+1}$ , converges Q-quadratically to  $(x_*, y_*)$  from any starting point  $(x_0, y_0)$  in  $\mathcal{X} \times \mathcal{Y}$ .

**Proof.** To obtain the first result, we observe that AW.1b implies that the Hessian of the Lagrangian function has bounded, Lipschitz continuous second derivatives in  $\Omega$ , since this is true for each individual component. Applying Theorem 3.1.6 (p. 29) to  $\nabla_x \ell(x, y)$  and  $c(x)$ , and using (15.2.3), we have that

$$\begin{aligned} 0 &= \nabla_x \ell(x_*, y_*) = \nabla_x \ell(x_k, y_k) + H_k(x_* - x_k) - A^T(x_k)(y_* - y_k) \\ &\quad + O(\|x_* - x_k\|^2) + O(\|x_* - x_k\|\|y_* - y_k\|) \\ &= \nabla_x \ell(x_k, y_k) + H_k s_k - A^T(x_k) s_k^y \\ &\quad + H_k(x_* - x_{k+1}) - A^T(x_k)(y_* - y_{k+1}) \\ &\quad + O(\|x_* - x_k\|^2) + O(\|x_* - x_k\|\|y_* - y_k\|) \\ &= H_k(x_* - x_{k+1}) - A^T(x_k)(y_* - y_{k+1}) \\ &\quad + O(\|x_* - x_k\|^2) + O(\|x_* - x_k\|\|y_* - y_k\|) \end{aligned}$$

and

$$\begin{aligned} 0 &= c(x_*) = c(x_k) + A(x_k)(x_* - x_k) + O(\|x_* - x_k\|^2) \\ &= c(x_k) + A(x_k)s_k + A(x_k)(x_* - x_{k+1}) + O(\|x_* - x_k\|^2) \\ &= A(x_k)(x_* - x_{k+1}) + O(\|x_* - x_k\|^2). \end{aligned}$$

Hence, combining the above,

$$\begin{aligned} &\begin{pmatrix} H_k & A^T(x_k) \\ A(x_k) & 0 \end{pmatrix} \begin{pmatrix} (x_* - x_{k+1}) \\ -(y_* - y_{k+1}) \end{pmatrix} \\ &= \begin{pmatrix} O(\|x_* - x_k\|^2) + O(\|x_* - x_k\|\|y_* - y_k\|) \\ O(\|x_* - x_k\|^2) \end{pmatrix}, \end{aligned} \quad (15.2.6)$$

from which it follows that

$$\|x_* - x_{k+1}\| = O(\|x_* - x_k\|) \max [O(\|x_* - x_k\|), O(\|y_* - y_k\|)], \quad (15.2.7)$$

since  $K(x_k, y_k)$  may be made arbitrarily close to  $K(x_*, y_*)$  by restricting  $\mathcal{X}$ , and  $K(x_*, y_*)$  is invertible under AO.3b. Thus, as  $\{y_k\}$  is assumed to converge, we may further restrict  $\mathcal{X}$  so that (15.2.7) implies that  $\{x_k\}$  converges to  $x_*$ , and (15.2.7) also implies that the rate is Q-superlinear. A Q-quadratic rate is apparent from (15.2.7) if  $\|y_k - y_*\| = O(\|x_k - x_*\|)$ .

The second result is established in essentially the same way. Equations (15.2.7) and

$$\|y_* - y_{k+1}\| = O(\|x_* - x_k\|) \max [O(\|x_* - x_k\|), O(\|y_* - y_k\|)]$$

follow, as before, from (15.2.6) by restricting  $\mathcal{X}$  and  $\mathcal{Y}$ , and thus further restrictions on these neighbourhoods imply that  $\{(x_k, y_k)\}$  converges to  $(x_*, y_*)$ . The required result follows immediately by applying Theorem 3.3.1 (p. 52) directly to the system in question.  $\square$

Notice that it does not necessarily follow that  $\{y_k\}$  converges Q-superlinearly to  $y_*$ . Conversely, there is actually no need for  $y_0$  to be accurate at all, so long as  $H_0$  and  $x_0$  are, since the multiplier  $y_1$  calculated from (15.2.5) is otherwise independent of  $y_0$ . For instance, if  $x_0 = x_*$ , (15.2.5) shows that  $s_0 = 0$  and  $y_1 = y_*$  irrespective of the choice of  $y_0$ . Moreover, there is no need for  $y_{k+1}$  to be chosen as the Lagrange multipliers from the subproblem (15.2.5) to obtain a Q-superlinear convergence rate. Often in practice, the least-squares multiplier estimates

$$y_k = y^{LS}(x_k) = \arg \min_y \|g(x_k) - A^T(x_k)y\|^2$$

are used; if  $A(x_*)$  is of full rank,

$$\|y_k - y_*\| = O(\|x_k - x_*\|),$$

as we saw in Lemma 14.2.2 (p. 578), and thus  $\{x_k\}$  converges Q-quadratically with such estimates.

The vast majority of SQP methods that have been proposed replace the Hessian of the Lagrangian  $\nabla_{xx}\ell(x_k, y_k)$  in (15.2.4a) or (15.2.5a) by a suitable symmetric “approximation”  $H_k$ . We say approximation very guardedly as it has been uncommon until recently for there to be any requirement that the two matrices are ever close. Most especially, there is no reason why  $\nabla_{xx}\ell(x_k, y_k)$  should ever be positive definite, while it was almost universally accepted that  $H_k$  should be. The main reasons for requiring that  $H_k$  be definite appear to have been pragmatic; a strictly convex quadratic program has at most one local minimizer, the SQP step is a descent direction for many popular merit functions (see Section 15.3), a number of available methods for solving quadratic programs required that the matrix be invertible (which is immediate if it is definite), and many of the most reliable quasi-Newton updating formulae generated positive-definite updates (see Section 8.4.1.2 [p. 281]). With hindsight, such a restriction merely reflected the limitations of the then-current quadratic programming technology, and we see little justification for such restrictions today, especially as exact second derivatives or sparse partitioned quasi-Newton approximations are relatively easy to obtain.

An SQP method is one that seeks to improve an estimate  $(x_k, y_k)$  of the solution to (15.2.2) by finding corrections  $(s_k, s_k^y)$  by solving one (or more) quadratic programming problems. The next estimate of the required solution will be

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + \alpha_k^x s_k \\ y_k + \alpha_k^y s_k^y \end{pmatrix}, \quad (15.2.8)$$

where the nonnegative stepsizes  $\alpha_k^x$  and  $\alpha_k^y$  may or may not be equal.

The prototypical *linesearch* SQP method finds  $s_k$  as the solution of the quadratic program (15.2.5). The stepsize  $\alpha_k^x$  is found by requiring that  $\phi(x_k + \alpha_k^x s_k)$  be sufficiently smaller than  $\phi(x_k)$  for some suitable merit function  $\phi(x)$ ; a merit function is a function against which progress towards a critical point may be measured—generally the smaller the function the better (see Section 15.3). This is achieved by performing a backtracking (Armijo) linesearch with a unit initial stepsize. Finally  $\alpha_k^y$  is set to either 1 or  $\alpha_k^x$ . The missing ingredients here are the choices of  $H_k$  and  $\phi$ , and it is mostly in these that the many proposed methods differ.

Our interest lies, of course, not with linesearch but with trust-region SQP methods. The ideas are by now, we hope, quite familiar: we impose an extra trust-region restriction on the model problem and dispense with the linesearch. Thus the prototypical SQP *trust-region* subproblem will be of the form

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H_k s \rangle + \langle g(x_k), s \rangle \quad (15.2.9a)$$

$$\text{subject to} \quad A(x_k)s + c(x_k) = 0 \quad (15.2.9b)$$

$$\text{and} \quad \|s\| \leq \Delta_k \quad (15.2.9c)$$

for some suitable trust-region radius  $\Delta_k$  and norm  $\|\cdot\|$ . The value of  $\alpha_k^x$  in (15.2.8) will depend only on how well the value of the merit function  $\phi(x)$  at  $x_k + s_k$  matches a model of the merit function at this point; good agreement will result in  $\alpha_k^x = 1$  and

possibly an increase in the trust-region radius for the next subproblem, while poor agreement will be punished with  $\alpha_k^x = 0$  and a reduction in the radius.

But now we start to see the difficulties that lie ahead. For instance, why should (15.2.9) have a solution? And, since we have not specified the merit function, what right have we to expect that (15.2.9), or indeed (15.2.4), is compatible with the merit function, and indeed in what sense is (15.2.9) a “model” of this function?

In fact, there is no reason at all why (15.2.9) has a solution. For a start, the linearized constraints may not be consistent; that is, there may be no value  $s$  for which  $A(x_k)s + c(x_k) = 0$ . For example, a linearization of the constraints  $x_1^2 + x_2^2 - 1 = 0$  and  $x_1 + x_2 - 1 = 0$  at  $(1, 1)^T$  leads to the inconsistent system

$$\begin{pmatrix} 2 & 2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

This can only happen if  $A(x_k)$  is rank deficient, but it is unrealistic to presume that it will never occur. However, while the reader may not consider this possibility likely, far more significantly, if we let

$$\theta_k = \min_{s \in \mathbb{R}^n} \|s\| \text{ subject to } A(x_k)s + c(x_k) = 0,$$

there can be no feasible point for (15.2.9) whenever  $\Delta_k < \theta_k$ . Thus, if the model gives a poor prediction of the merit function—for whatever reason—we cannot rely on our usual mechanism of simply reducing the trust-region radius to make further progress. This is illustrated in Figure 15.2.1.

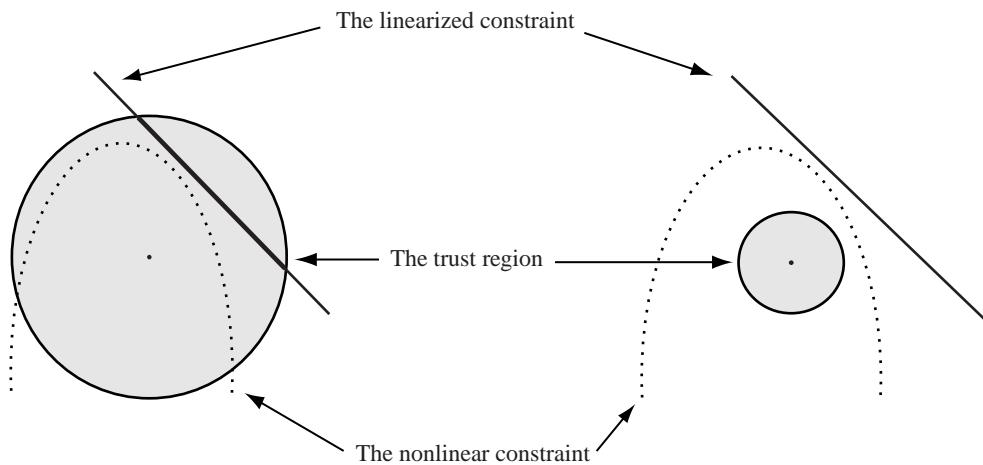


Figure 15.2.1: The intersection between the linearization of a nonlinear constraint and a spherical trust region. In the left figure, the trust-region radius is sufficiently large for the trust region and the linearized constraint to intersect. This is not so for the smaller trust region illustrated in the right figure.

It is for these reasons that the quadratic program (15.2.9) is rarely the appropriate subproblem for trust-region SQP methods. This and other related issues will form the bulk of this chapter.

### Notes and References for Subsection 15.2.1

A thorough treatment of the history, theory, and practice of SQP methods is given by Boggs and Tolle (1995). The first SQP method was proposed by Wilson (1963), while the first semblance of an SQP trust-region method appears in Beale (1967) and Sargent (1974). Most line-search SQP methods follow that proposed by Pschenichny (1970), while, because of the difficulties hinted at above, there is a larger variation in possible trust-region SQP methods.

As we have seen, the first-order optimality conditions for

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, Hs \rangle + \langle \nabla_x \ell(x, y), s \rangle \quad (15.2.10a)$$

$$\text{subject to} \quad A(x)s + c(x) = 0 \quad (15.2.10b)$$

are that

$$\begin{pmatrix} H & A^T(x) \\ A(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ -s^y \end{pmatrix} = - \begin{pmatrix} \nabla_x \ell(x, y) \\ c(x) \end{pmatrix}. \quad (15.2.11)$$

Assuming that  $A(x)$  is of full rank, and letting  $R(x)$  and  $N(x)$  be matrices whose columns span, respectively, the range-space and null-space of  $A(x)$ , we may write  $s = R(x)s_r + N(x)s_n$ . On substituting into (15.2.11),  $s_r$  is completely determined by the constraints, as the solution to the nonsingular system

$$A(x)R(x)s_r = -c(x), \quad (15.2.12)$$

while  $s_n$  then satisfies

$$H_{nn}s_n = -N^T(x)\nabla_x \ell(x, y) - H_{nr}s_r, \quad (15.2.13)$$

where  $H_{nn} = N^T(x)HN(x)$  and  $H_{nr} = N^T(x)HR(x)$ . Returning to (15.2.10) and performing the same substitution for  $s$  also yields (15.2.13), but the further requirement in (15.2.10) that a minimizer be sought suggests that the reduced Hessian  $H_{nn}$  should be positive semidefinite; to make this solution unique, the requirement is normally strengthened to insist that the reduced Hessian be positive definite.

Early SQP methods assumed that  $H$  itself was positive definite. This is clearly stronger than requiring that  $H_{nn}$  be definite. Most importantly, second-order optimality conditions for the original problem suggest that the reduced Hessian of the Lagrangian should be at least positive semidefinite, but that there is no reason for the Hessian of the Lagrangian itself to be definite. Advocates of this assumption cite simplicity and were clearly keen to define  $H$  via one of the positive definite quasi-Newton updating formulae which had proven so successful in unconstrained optimization. However, in our opinion, the contortions that were necessary to bend the quasi-Newton updates into a suitable form (see, for example, Powell, 1978) underline the difficulties with the approach. In mitigation, when inequality constraints are introduced, the dimension of  $N(x)$  may change dramatically from one iteration to the next, and it is then certainly convenient that  $H$  is positive definite. Remarkably, the very first SQP methods (Wilson, 1963; Beale, 1967) used the exact Hessian of the Lagrangian, but until recently very few authors considered this choice (see Boggs, Tolle, and Kearsley, 1994; Boggs, Kearsley, and Tolle, 1999; and also Bonnans and Launay, 1995, who sometimes modify the exact Hessian).

More recent methods have aimed at ensuring that  $H_{nn}$  is positive definite using positive definite secant updates. However, this leads one to wonder how to handle the other matrices  $H_{nr}$  and  $H_{rr} = R^T(x)HR(x)$ , and it is here that most of the current proposals vary. Murray and Wright (1978) suggested that  $H_{nr}$  and  $H_{rr}$  should be set to zero. This gives what is known as a *reduced Hessian* method. With an appropriate secant update formula, such a scheme is a two-step Q-superlinearly convergent method so long as  $\alpha_x = 1$  (Nocedal and Overton, 1985). A related reduced Hessian method, due to Coleman and Conn (1982a), replaces (15.2.12) by

$$A(x)R(x)s_r = -c(x + N(x)s_n) \quad (15.2.14)$$

in the vicinity of a stationary point, or  $s_r = 0$  elsewhere. This method is also two-step Q-superlinearly convergent. Perhaps more surprisingly, Byrd (1990) shows that the iterates  $x + N(x)s_n$  have in fact a one-step Q-superlinear rate and that such a rate is common for many SQP methods that involve the correction (15.2.14). Byrd and Nocedal (1991) show that these local results are not affected by global convergence concerns (see Section 15.3) provided that precautions are taken. Another possibility with the same theoretical convergence properties, proposed by Gilbert (1991), is to maintain a secant approximation to the inverse of  $H_{nn}$ . A similar convergence rate is also achieved by methods that use Broyden-type secant methods to approximate the rectangular matrix  $(H_{nn} \ H_{nr})$  (see Nocedal and Overton, 1985; or Fontecilla, Steihaug, and Tapia, 1987). Gurwitz (1994) prefers updates that treat the portions  $H_{nn}$  and  $H_{nr}$  separately while maintaining a positive definite approximation to the former. Coleman and Fenyes (1992) propose a similar method, and also a second method that additionally maintains an approximation to  $H_{rr}$ . These methods appear to perform slightly better than those that only maintain a nonzero  $H_{nn}$ . Finally, Biegler, Nocedal, and Schmid (1995) note that (15.2.13) does not actually require  $H_{nr}$  but rather the vector  $H_{nr}s_r$ , and thus they propose approximating this term directly by either finite differences or by a Broyden update.

What can we say about the Lagrange multiplier estimates? The values  $s^y$  from (15.2.11) satisfy

$$R^T(x)A^T(x)s^y = R^T(x)\nabla_x\ell(x, y) + H_{nr}^T s_n + H_{rr} s_r.$$

Clearly, setting  $H_{nr}$  and  $H_{rr}$  to zero implies that  $y + s^y$  are least-squares multiplier estimates  $y^{LS}(x)$  evaluated at  $x$ . If these neglected terms are included,  $y + s^y$  are approximations to least-squares multiplier estimates at  $x + s$ . Thus, rather than use these approximations, many authors prefer to use the current least-squares estimates  $y^{LS}(x + s)$ , for which

$$R^T(x + s)A^T(x + s)y^{LS}(x + s) = R^T(x + s)g(x + s), \quad (15.2.15)$$

directly.

Finally, although we have argued that maintaining a positive definite approximation to the Hessian of the Lagrangian function is unnecessarily restrictive, it is more reasonable when approximating the Hessian of the augmented Lagrangian. Indeed, if  $\sigma$  is a scalar, we can add the term  $\sigma\|A(x)s + c(x)\|_2^2$  to the objective function of (15.2.10) without changing its solution. But the Hessian of this modified problem is  $H + \sigma A^T(x)A(x)$ , which can be expected to be positive definite for sufficiently large  $\sigma$ . Such a method was first proposed by Tapia (1977), while suitable secant update formulae for  $H + \sigma A^T(x)A(x)$  are discussed by Byrd, Tapia, and Zhang (1992). Gould and Toint (2000) survey the recent developments in SQP along the lines developed in this chapter.

### 15.2.2 Methods for Inequality Constraints

Nonlinear programming problems rarely exclusively involve equality constraints, but typically involve a mixture of equations and inequalities. We thus turn to the general problem (15.1.1). Theorem 3.2.4 (p. 40) shows that the first-order optimality conditions for this problem are that

$$\nabla_x \ell(x, y) = 0, \quad c_{\mathcal{E}}(x) = 0, \quad c_{\mathcal{I}}(x) \geq 0, \quad y_{\mathcal{I}} \geq 0, \quad \text{and} \quad \langle c_{\mathcal{I}}(x), y_{\mathcal{I}} \rangle = 0,$$

where, as usual, the Lagrangian function

$$\ell(x, y) = f(x) - \langle c(x), y \rangle,$$

$c^T(x) = (c_{\mathcal{E}}^T(x) \ c_{\mathcal{I}}^T(x))$ , and  $y^T = (y_{\mathcal{E}}^T \ y_{\mathcal{I}}^T)$ , and we have written  $c_{\mathcal{E}}(x)$  and  $c_{\mathcal{I}}(x)$  for the vectors of values of the equality and inequality constraints, and  $y_{\mathcal{E}}$  and  $y_{\mathcal{I}}$  for their corresponding Lagrange multipliers.

Until recently, most algorithms for (15.1.1) were primarily of the active or working set varieties. As we have already mentioned, a working set method is one that aims to solve (15.1.1) by predicting which of the inequalities will be active (i.e., which of the  $c_i(x) = 0$ ) and which are inactive (i.e.,  $c_i(x) > 0$ ) at the solution. Once these sets are known, the problem can be solved as if it involves only equality constraints, namely, the genuine equalities and those inequalities deemed to be active at the solution. The main justification, therefore, for much of the work described in Section 15.2.1 is as a tool for analysing working set methods.

The principal differences between active set methods is in the way that the active set is assigned. In inequality-(constrained) sequential quadratic programming (IQP) methods, no a priori choice of the active set is made when choosing the correction  $s$ ; rather,  $s$  is obtained by solving the quadratic programming problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H_k s \rangle + \langle g(x_k), s \rangle \quad (15.2.16a)$$

$$\text{subject to} \quad A_{\mathcal{E}}(x_k)s + c_{\mathcal{E}}(x_k) = 0 \quad \text{and} \quad A_{\mathcal{I}}(x_k)s + c_{\mathcal{I}}(x_k) \geq 0, \quad (15.2.16b)$$

which includes linear approximations of all of the constraints; for trust-region IQP methods, the trust-region constraint (15.2.9c) will also be involved. The active set for this problem is taken as a prediction of that for (15.1.1). In equality-(constrained) sequential quadratic programming (EQP) methods, the active set is assigned prior to the selection of  $s$  (primarily on the basis of inequalities that are close to being active and whose Lagrange multiplier estimates are positive), and  $s$  is found directly by solving (15.2.4) subject to the linear equality constraints  $A(x)_{\mathcal{A}}s + c_{\mathcal{A}}(x) = 0$ , where the subscript  $\mathcal{A}$  denotes those constraints that are considered to be active. Which strategy is preferable is a matter for some debate.

We may summarize the main local convergence results as follows.

**Theorem 15.2.2** Suppose that AW.1b holds in some neighbourhood  $\Omega$  of the first-order critical point  $(x_*, y_*)$  of (15.1.1), and that AO.1b, AO.3, and AO.4 hold there. Then, the following is true.

- (i) Consider any  $\{y_k\}$  converging to  $y_*$ . Then there is a neighbourhood  $\mathcal{X} \subset \Omega$  of  $x_*$  for which the sequence  $\{x_k\}$ , generated by the iteration  $x_{k+1} = x_k + s_k$ , where  $s_k$  is the local minimizer of (15.2.16) closest to the origin, converges Q-superlinearly to  $x_*$  from any starting point  $x_0$  in  $\mathcal{X}$ . Furthermore, if  $\|y_k - y_*\| = O(\|x_k - x_*\|)$ , the convergence is Q-quadratic.
- (ii) Let  $\{x_k\}$  and  $\{s_k\}$  be defined as in (i), with  $\{y_{k+1}\}$  being the Lagrange multipliers associated with  $\{s_k\}$ . Then there is a neighbourhood  $\mathcal{X} \subset \Omega$  of  $x_*$  and another neighbourhood  $\mathcal{Y}$  of  $y_*$  for which the sequence  $\{(x_k, y_k)\}$  converges Q-quadratically to  $(x_*, y_*)$  from any starting point  $(x_0, y_0)$  in  $\mathcal{X} \times \mathcal{Y}$ .
- (iii) In either case, the set of constraints that are active at  $x_*$  are precisely those that are active for the subproblem (15.2.16) at  $s_k$  for large enough  $k$ .

**Proof.** Consider the quadratic program

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && \frac{1}{2} \langle s, H(x_* + p, y_* + q)s \rangle + \langle g(x_* + p), s \rangle \\ & \text{subject to} && A_{\mathcal{E}}(x_* + p)s + c_{\mathcal{E}}(x_* + p) = 0 \\ & \text{and} && A_{\mathcal{T}}(x_* + p)s + c_{\mathcal{T}}(x_* + p) \geq 0, \end{aligned} \tag{15.2.17}$$

where  $p$  and  $q$  are parameters. Clearly  $s = 0$  and  $y_*$  satisfies conditions AO.1b, AO.3, and AO.4 of Theorem 3.2.7 (p. 42) for the quadratic program (15.2.17) when  $(p, q) = 0$ . Now let  $(p, q) = (x_k - x_*, y_k - y_*)$ . In this case, (15.2.17) is simply (15.2.16). But Theorem 3.2.7 shows that  $s_k$  is the only local minimizer near, and thus the closest to, the origin (and its vector of Lagrange multipliers is unique) provided  $(x_k, y_k)$  are sufficiently close to  $(x_*, y_*)$ ; and it also shows that conditions AO.1b, AO.3, and AO.4 continue to apply there. More importantly,  $\mathcal{A}(x_k) = \mathcal{A}(x_*)$  for all sufficiently close  $(x_k, y_k)$ . The remaining results now follow immediately from Theorem 15.2.1, as the subproblem (15.2.16) may be considered simply as if the inactive constraints had been discarded and the remainder considered as equalities.

□

## Notes and References for Subsection 15.2.2

An assessment of the advantages and disadvantages of IQP and EQP methods is made by Murray and Wright (1982).

Theorem 15.2.2 is due to Robinson (1974). The important assumption here is that  $A_{\mathcal{A}(x_*)}(x_*)$  is of full rank (AO.1b), since this ensures that the Lagrange multipliers at  $x_*$ , as well as those for (15.2.16) for sufficiently large  $k$ , are unique. If  $A_{\mathcal{A}(x_*)}(x_*)$  is not of full rank, the limiting multipliers may not be unique, and the SQP method using the estimates ob-

tained from (15.2.16) may not converge Q-quadratically. Of course such an assumption on the rank of  $A_{\mathcal{A}(x_*)}(x_*)$  is a relatively strong first-order constraint qualification, and S. J. Wright (1999) shows that it is possible to replace this assumption by a weaker one due to Mangasarian and Fromovitz (1967) while still obtaining Q-quadratic convergence. To do so, the SQP subproblem must be modified slightly to ensure that its Lagrange multipliers are (locally) unique. In fact, Wright's subproblem is equivalent to that which would arise if the augmented Lagrangian function (14.3.34) (p. 593) for (15.2.16) were minimized with respect to  $x$  and simultaneously maximized with respect to  $y$  while ensuring that  $y_T \geq 0$ . To ensure a Q-quadratic rate, the parameter  $\mu$  in (14.3.34) must approach zero as  $O((x_k - x_*, y_k - y_*))$ . Hager (1999a) shows that it is also possible to remove the strict complementary-slackness requirement (3.2.14) (p. 42) so long as the second-order sufficiency condition (3.2.13) (p. 42) is strengthened. See also Bonnans and Launay (1992).

A disadvantage of many of the methods we have reviewed so far is that they may be inappropriate if the number of variables is large. In particular, unless function values are expensive, the dominant cost of the methods tends to be in solving linear systems; for inequality problems, this may be particularly acute as each subproblem may require the solution of a sequence of such systems. If  $n$  is large, there is little hope unless the required systems are either small or sparse. There are two important cases where this is so. Firstly, if the number of equality or active constraints is close to  $n$ , reduced Hessian methods, such as those proposed by Coleman and Conn (1982a), Gilbert (1991), and Biegler, Nocedal, and Schmid (1995), which maintain the matrix  $H_{nn}$  but ignore  $H_{nr}$  and  $H_{rr}$ , may be successful. The only systems that need to be solved involve  $H_{nn}$  (small) and  $A(x)R(x)$  and its transpose (sparse, we hope). See also Zhang and Zhu (1999). Secondly, if the matrix  $H$  is sparse, sparse methods for linear systems (when an EQP method is used) or quadratic programming (for IQP methods) may be employed. This will often be the case if  $H$  is chosen as the Hessian of the Lagrangian function, or from a structured or sparse quasi-Newton updating formula (see, for example, Toint, 1977; Griewank and Toint, 1982a; Conn, Gould, and Toint, 1990; and Fletcher, 1995).

One of the main advances in methods for the unconstrained minimization of large problems was the recognition that a Newton-like direction need not be computed very accurately when far from a stationary point (see Dembo, Eisenstat, and Steihaug, 1982). Clearly, when equality constraints are present a similar result would be valuable, but there has been remarkably little work on this topic. For equality-constrained problems involving relatively few constraints, Fontecilla (1990) notes that (15.2.12) and (15.2.15) are small systems, and the only large system is (15.2.13). He thus proposes a method that initially solves (15.2.13) to low accuracy. Alas, for fast asymptotic convergence, (15.2.13) must eventually be solved to high accuracy, which limits the effectiveness of this proposal. When the constraints are inequalities and an active set IQP method is used, Murray and Prieto (1995) show that it is possible to stop the solution of the quadratic programming subproblem at the first stationary point encountered rather than solving the problem to completion, but unfortunately this may still be rather expensive.

### 15.2.3 Quadratic Programming

Of course, the cornerstone of any SQP algorithm will be the underlying quadratic programming method. Modern quadratic programming methods fall broadly into two

categories: active or working set methods and interior-point methods. We encountered the former in Section 7.8 for the special case where the linear constraints are simple bounds. Working set methods for the more general problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, Hs \rangle + \langle g, s \rangle \quad (15.2.18a)$$

$$\text{subject to} \quad A_{\mathcal{E}}s + c_{\mathcal{E}} = 0 \quad \text{and} \quad A_{\mathcal{I}}s + c_{\mathcal{I}} \geq 0 \quad (15.2.18b)$$

proceed in essentially the same way. At any stage, the working set  $\mathcal{W}$  is chosen as the intersection of the equality constraints  $\mathcal{E}$  and an appropriate subset of the inequality constraints  $\mathcal{I}$ . If the solution to the subproblem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, Hs \rangle + \langle g, s \rangle \quad (15.2.19a)$$

$$\text{subject to} \quad A_{\mathcal{W}}s + c_{\mathcal{W}} = 0 \quad (15.2.19b)$$

is infeasible with respect to the remaining constraints  $\mathcal{R} = \mathcal{I} \setminus (\mathcal{W} \cap \mathcal{E})$ , it cannot solve (15.2.18). Typically, in this case one or more elements of  $\mathcal{R}$  that are infeasible at the solution to (15.2.19) are transferred to  $\mathcal{W}$ , and the subproblem (15.2.19) is resolved. If, on the other hand, the solution  $x_{\mathcal{W}}$  to (15.2.19) satisfies (15.2.18b), it is a candidate solution for (15.2.18). To see if this is actually the solution, we need to compute Lagrange multipliers  $y_{\mathcal{W}}$  for which

$$\begin{pmatrix} H & A_{\mathcal{W}}^T \\ A_{\mathcal{W}} & 0 \end{pmatrix} \begin{pmatrix} x_{\mathcal{W}} \\ -y_{\mathcal{W}} \end{pmatrix} = - \begin{pmatrix} g \\ c_{\mathcal{W}} \end{pmatrix}; \quad (15.2.20)$$

cf. (4.4.8) and (4.4.14). If all the components  $y_{\mathcal{W} \cap \mathcal{I}} \geq 0$ , Theorem 3.2.4 (p. 40) shows that  $x_{\mathcal{W}}$  is a first-order critical point for the problem; whether this is a local minimizer depends upon the definiteness of  $H$  on an appropriate cone (see Theorem 3.2.5 [p. 41] and recall that the second-order necessary optimality conditions are sufficient for quadratic programs), which is hard to check except in the convex case, where  $H$  is positive semidefinite. If there are negative Lagrange multipliers, progress may be made by deleting one (and, with care, sometimes more) of the corresponding inequality constraints from  $\mathcal{W}$  and resolving the resulting subproblem. Convergence is ensured in a finite number of iterations so long as the progress is made at each step. This is guaranteed so long as constraints in  $\mathcal{R}$  are strictly inactive at each stage, and may be ensured in all cases using so-called anticycling rules. The actual number of steps may be exponential as a function of the dimension of the problem in the worst, but fortunately atypical, case. The dominant work per iteration is in solving a system like (15.2.20), although slight complications occur in the nonconvex case since then the solution to (15.2.20) may give a saddle point and not a minimizer of (15.2.19). The cost of the iteration is generally modest, as the working sets for consecutive subproblems are closely related, and this enables us to update rather than recompute the factors of matrices required to solve (15.2.19).

Although the general quadratic programming problem is NP hard, the convex problem is not. Thus the working set method we have just considered is certainly inefficient

from an algorithmic complexity perspective. This is not the case with variants of the interior-point or barrier methods we considered in Chapter 13. We will not consider such methods further here, but simply note that the fact that the objective is quadratic enables a number of efficiencies to be made. We believe that research in this area is likely to be extremely active over the next few years.

We note that it is unlikely that either of these approaches is sufficient on its own. When we are far from the solution, we would expect to have little idea of the correct active set. In this case, we would probably expect an interior-point method to be more effective than a working space one, simply because the combinatorial problem is handled better by the former. On the other hand, when an SQP method is close to the solution, Theorem 15.2.2 indicates that the active set from the previous subproblem is likely to be an excellent prediction for the initial working set for the current one. In this case, a working set method would be expected to be the most effective means of solving the current subproblem. By contrast, interior-point methods are poorly adapted to coping with information about the correct active set, rather preferring to find this information anew. For these reasons, we suggest that it may be necessary to include *two* quadratic programming codes—one working set and one interior-point-based—in any SQP package for large-scale nonlinear programming.

### Notes and References for Subsection 15.2.3

There are a large number of closely related active/working set quadratic programming methods. Most existing algorithms for convex quadratic programming (see, for example, Frank and Wolfe, 1956; Wolfe, 1959; Van de Panne and Whinston, 1969; Bartels, Golub, and Saunders, 1970; Murray, 1971a; Best and Ritter, 1976; Powell, 1981b, 1985; Goldfarb and Idnani, 1983; Gill et al., 1984) are theoretically identical—in exact arithmetic they all generate the same sequence of iterates from a given starting point—provided that they operate under identical rules for choosing which constraints to add to and remove from the working set (see Best, 1984; Djang, 1979; and Pang, 1981). They differ merely in the algebraic methods for solving (15.2.20). The same is true of methods for the general problem given by Fletcher (1971a), Keller (1973), Gill and Murray (1978), and Bunch and Kaufman (1980), and those for the general sparse problem by Gill et al. (1990, 1991) and Gould (1991). Anticycling rules specifically for quadratic programming have been given by Chang and Cottle (1980) and Gould (1991), although, as Osborne (1985) pointed out, in principle many of the rules developed for linear programming are equally appropriate. See also the surveys by Fletcher (1987a, Chapter 10, 1987b), the bibliography in Cottle, Pang, and Stone (1992), and the book by Boot (1964). Active/working set methods for quadratic programming are available in the Harwell (2000), IMSL (1999), and NAG (1998) libraries; the former includes VE09, which is designed for large, nonconvex problems.

Of course, linear programming is a special case of convex quadratic programming. The most famous working set method in this case is the simplex method described in detail by Dantzig (1963). There are hundreds of extensions to, and applications of, this famous method, and we have no space to do justice to them here. We merely mention that the book by Chvátal (1983) is our favourite introduction to this fascinating subject and that numerical aspects of the method are covered by Murtagh (1981).

Special interior-point methods with excellent complexity bounds have been proposed for the convex case by Ye and Tse (1989), Monteiro and Adler (1989), Ye (1992), Carpenter et al. (1993), Goldfarb and Liu (1993), Sun (1993), and Monteiro and Tsuchiya (1998), while a method that finds an approximation to the global solution in the nonconvex case is given by Vavasis (1992a). Algorithms that aim for first-order critical points in the general case—without, of course, guaranteed polynomial complexity—are considered by Vanderbei (1994), Boggs, Domich, and Rogers (1995), Vanderbei and Shanno (1997), and Conn, Gould, Orban, and Toint (2000), and there is evidence that such techniques are indeed able to solve very large problems. Interior-point methods lie at the heart of the LOQO (see Vanderbei, 1994; Vanderbei and Shanno, 1997; and Shanno, 1999), CPLEX 6.0 (1998), and **VE12** (see Conn, Gould, Orban, and Toint, 2000) software packages for large-scale quadratic programming.

Methods for “warm-starting” working set-based quadratic programming methods when solving a sequence of closely related problems are discussed by Gill, Murray, Saunders, and Wright (1985, 1990).

### 15.3 Merit Functions and SQP Methods

In Section 14.1, we introduced the idea of a penalty function as a means of balancing the sometimes conflicting goals of maintaining feasibility and reducing the objective function. Penalty functions may be used in two ways. Firstly, a suitable direction that maintains this balance may be constructed by considering the local behaviour of the penalty function—this is the approach we considered in Chapter 14. By contrast, we might equally well use the penalty function as a means to assess the quality of a predetermined search direction. In particular, one might believe that a step along a given search direction is “good” if the penalty function decreases, and “bad” otherwise. In this case, the penalty function is being used as a merit function to ensure convergence of a basic iteration from arbitrary starting points.

The crucial property of a (first-order) *merit* function is that it should have (first-order) critical points at (first-order) critical points of the underlying problem (at least in some limiting sense if the merit function depends upon parameters). We saw that this was so for the smooth penalty functions we considered in Chapter 14, provided the parameters are suitably adjusted. A highly desirable additional property is that these critical points should coincide, that is, that the merit function should not have extra critical points at values for which the underlying problem is not critical. Unfortunately, this is not the case for the functions of Chapter 14, which may have additional critical points when the constraint Jacobian is rank deficient. Even if the Jacobian is not rank deficient, the penalty function can have nonfeasible critical points.

Ideally there should be an intimate connection between the merit function and the model problem which defines the search direction. For example, this is clearly the case with unconstrained minimization, where the objective function serves as the merit function, and a Newton-like model of the merit function gives rise to search directions that are useful in their own right, indeed the very directions that would have been chosen without consideration of the merit function.

In this section, we shall consider a number of merit functions that are appropriate for SQP methods. We should inject a note of caution here. We do not mean that these functions are necessarily appropriate when the search direction is chosen by solving the particular quadratic program (15.2.5) or (15.2.9), merely that they are appropriate for *specific* related quadratic programs. That is, the merit function often goes hand-in-hand with its own special model quadratic program.

### 15.3.1 The Augmented Lagrangian Penalty Function

In Sections 14.2–14.4, we considered the augmented Lagrangian penalty function

$$\Phi_y(x, \mu) = f(x) - \langle c(x), y \rangle + \frac{1}{2\mu} \|c(x)\|_2^2. \quad (15.3.1)$$

The standard Newton model for this function at  $x$  is

$$\begin{aligned} m(x, s, \mu) &= \Phi_y(x, \mu) + \langle \nabla_x \Phi_y(x, \mu), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} \Phi_y(x, \mu) s \rangle \\ &= \Phi_y(x, \mu) + \langle \nabla_x \ell(x, y(x, \mu)), s \rangle \\ &\quad + \frac{1}{2} \left\langle s, \left( \nabla_{xx} \ell(x, y(x, \mu)) + \frac{1}{\mu} A^T(x) A(x) \right) s \right\rangle \\ &= \ell(x, y) + \langle \nabla_x \ell(x, y), s \rangle \\ &\quad + \frac{1}{2} \langle s, \nabla_{xx} \ell(x, y(x, \mu)) s \rangle + \frac{1}{2\mu} \|A(x)s + c(x)\|_2^2, \end{aligned}$$

where  $y(x, \mu) = y - c(x)/\mu$ . On defining

$$v = \frac{1}{\mu} (c(x) + A(x)s),$$

the trust-region subproblem is simply to

$$\begin{array}{ll} \underset{s, v}{\text{minimize}} & \langle \nabla_x \ell(x, y), s \rangle + \frac{1}{2} \langle s, \nabla_{xx} \ell(x, y(x, \mu)) s \rangle + \frac{1}{2} \mu \|v\|_2^2 \\ \text{subject to} & A(x)s + c(x) - \mu v = 0 \quad \text{and} \quad \|s\| \leq \Delta. \end{array} \quad (15.3.2)$$

Thus, in view of (15.3.2), the standard trust-region subproblem for (15.3.1) may be formulated as a quadratic program and thus interpreted as an SQP method. Notice that the advantages here are that, unlike (15.2.9), the subproblem (15.3.2) always has a feasible solution and that if  $\mu$  converges to zero, the two subproblems coincide.

### Notes and References for Subsection 15.3.1

See Murray (1969), Biggs (1972), and Bartholomew-Biggs (1987) for further details.

### 15.3.2 Nonsmooth Exact Penalty Functions

In Section 14.5, we showed that functions of the form

$$\phi(x, \sigma) = f(x) + \sigma \|c(x)^{\mathcal{T}-}\| \quad (15.3.3)$$

are nonsmooth exact penalty functions for the general problem (15.1.1) so long as  $\sigma$  is sufficiently large. While the nonsmoothness of such functions might, at first sight, appear to be a handicap when building suitable minimization algorithms, we also derived powerful, general-purpose methods for just such a task in Chapter 11. The algorithm we developed in Section 11.1 hinged crucially on approximately minimizing an appropriate model of the nonsmooth objective function, but we also saw in Section 11.5 that suitable models give rise to linear or quadratic programming models when polyhedral, convex norms are used both in (15.3.3) and to define the trust-region constraint  $\|s\| \leq \Delta$ . Thus, in principle, we already have all the ingredients for a globally convergent SQP method. In this section we assemble all of these ingredients into just such a method.

### 15.3.2.1 Global Convergence

Perhaps the most useful model for our purpose is the second-order approximation

$$m(x, H, s) = f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, Hs \rangle + \sigma \| (c(x) + A(x)s)^{\mathcal{I}^-} \| \quad (15.3.4)$$

to (15.3.3). Such a model is suitable for Algorithm 11.1.1 (p. 413) because of Theorem 11.5.1 (p. 436), and this algorithm, as well as the nonmonotone variant discussed in Section 11.3.1, are globally convergent because of Theorem 11.2.5 (p. 420) (or, in the second case, Theorem 11.3.1 [p. 423]). It is, of course, necessary to ensure that  $\sigma$  is sufficiently large that critical points of (15.3.3) are likely to coincide with those for (15.1.1). We have already discussed how one might achieve this at the end of Section 14.5. Formally, the following assumptions are required.

**AI.2** The sequence  $\{x_k\}$  generated by Algorithm 11.1.1, (or its nonmonotone counterpart) applied to the minimization of (15.3.3), using the model (15.3.4) and with steps chosen to satisfy AA.1n, has a limit point  $x_*$  for which  $c(x_*)^{\mathcal{I}^-} = 0$ .

**AW.4** The penalty parameter satisfies

$$\sigma > \min_{y \in \mathcal{Y}(x_*)} \|y\|_{\mathbb{D}},$$

where  $\mathcal{Y}(x_*)$  is the set of all Lagrange multipliers at  $x_*$ .

**AM.4j** The sequence of second derivative approximations  $\{H_k\}$  is bounded.

We then have our central global convergence result.

**Theorem 15.3.1** Suppose that AW.1b, AW.4, AM.4j, and AI.2 hold. Then  $x_*$  is a first-order critical point for (15.1.1).

**Proof.** Assumption AM.4j is sufficient for Theorem 11.5.1 (p. 436) to ensure that the model (15.3.4) satisfies the assumptions of Theorem 11.2.5 (p. 420), and thus that  $x_*$  is a first-order critical point for (15.3.3). The remaining assumptions and Theorem 14.5.1 (p. 610) then imply the required result.  $\square$

It is worth stressing that Theorem 15.3.1, unlike many of the other global convergence results for constrained optimization (see Chapter 14), makes no full-rank assumption on the Jacobian of the constraints active at  $x_*$ .

As we saw in Section 11.5, approximately solving the subproblem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad m(x, H, s) \quad \text{subject to} \quad \|s\| \leq \Delta \quad (15.3.5)$$

is particularly appealing when the norms involved are polyhedral, as then the subproblem is a quadratic program. It is traditional to use an  $\ell_\infty$  norm for the trust region. Choosing  $\|\cdot\| = \|\cdot\|_1$  in (15.3.3) and (15.3.4) gives rise to what is commonly known as the  $S\ell_1$ QP method, while, by analogy,  $\|\cdot\| = \|\cdot\|_\infty$  gives the  $S\ell_\infty$ QP method.

### 15.3.2.2 Asymptotic Convergence of the Basic Method

While the global convergence of Algorithm 11.1.1 (p. 413) and its nonmonotone counterpart do not require that we solve the subproblem (15.3.5) exactly at each iteration, it is interesting to consider the consequences of doing so. Recall that the idealized SQP method is based not on solving (15.3.5) but instead on solving

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, Hs \rangle + \langle g(x), s \rangle \quad (15.3.6a)$$

$$\text{subject to} \quad A_{\mathcal{E}}(x)s + c_{\mathcal{E}}(x) = 0, \quad A_{\mathcal{I}}(x)s + c_{\mathcal{I}}(x) \geq 0, \quad (15.3.6b)$$

$$\text{and} \quad \|s\| \leq \Delta. \quad (15.3.6c)$$

As we shall now see, under certain circumstances the two problems actually yield the same critical points.

**Theorem 15.3.2** Suppose that  $s$  is a first-order critical point for (15.3.6) with an associated set of Lagrange multipliers  $y$ . Then  $s$  is also a first-order critical point for (15.3.5) provided that  $\sigma \geq \|y\|_{\mathcal{D}}$ . The reverse is true so long as  $s$  is feasible for (15.3.6). If, in addition,  $\|s\| < \Delta$  and  $\sigma > \|y\|_{\mathcal{D}}$ , the second-order sufficiency conditions for the two problems are equivalent.

**Proof.** The first-order criticality conditions implied by Theorem 3.2.15 (p. 49) for (15.3.6) are that  $x$ ,  $y$ , and  $u \in \partial\|x\|$  satisfy the primal feasibility conditions

$$A_{\mathcal{E}}(x)s + c_{\mathcal{E}}(x) = 0, \quad A_{\mathcal{I}}(x)s + c_{\mathcal{I}}(x) \geq 0, \quad \text{and} \quad \|s\| \leq \Delta, \quad (15.3.7)$$

the dual feasibility conditions

$$Hs + g(x) - A^T(x)y + uz = 0, \quad y_{\mathcal{I}} \geq 0, \quad \text{and} \quad z \geq 0,$$

and the complementary slackness conditions

$$\langle A(x)s + c(x), y \rangle = 0 \quad \text{and} \quad (\|s\| - \Delta)z = 0. \quad (15.3.8)$$

On letting  $y^m = -y/\sigma$  and observing that  $\|(c(x) + A(x)s)^{\mathcal{T}-}\| = 0$ , it follows immediately from Corollary 11.4.2 (p. 430) that  $y^m \in \partial\|(c(x) + A(x)s)^{\mathcal{T}-}\|$ , and the remaining first-order criticality conditions for (15.3.5) required by Theorem 3.2.15 (p. 49) are contained in (15.3.7)–(15.3.8).

The equivalence of the second-order sufficiency conditions for the two problems when  $s$  is feasible,  $\|s\| < \Delta$  and  $\sigma > \|y\|_D$ , follows from Theorem 14.5.1 (p. 610) since (15.3.4) is an exact penalty function for (15.3.6).  $\square$

It is actually possible to show that the second-order sufficiency conditions for the two problems are equivalent without requiring  $\|s\| < \Delta$ , but this will not be necessary in what follows.

One significant advantage of using the subproblem (15.3.5) rather than (15.3.6) is that the former always has feasible points while, as we have seen in Section 15.2.1, the latter may not. Although we have stressed that it is not necessary to find the model minimizer, we shall now investigate this particular choice by assuming that

**AA.1c** the step  $s$  is chosen as a local minimizer of the model (15.3.5) for which AA.1n holds.

Of course the (global) model minimizer satisfies AA.1c. In practice, one might switch to minimizing the model when one believes that a suitable neighbourhood of a critical point has been found. We shall also strengthen AI.2 by assuming that

**AI.2b** the sequence  $\{x_k\}$  generated by Algorithm 11.1.1 (p. 413) (or its non-monotone counterpart) applied to the minimization of (15.3.3) with steps chosen to satisfy AA.1c has a single limit point  $x_*$  for which  $c(x_*)^{\mathcal{T}-} = 0$ .

To say more, we shall now assume that AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$ ; these assumptions imply that there is a unique  $y_*$  at  $x_*$ , and consequently AW.4 is simply that  $\sigma > \|y_*\|_D$ . In this case, Theorems 3.2.14 (p. 49) and 14.5.1 (p. 610) show that  $x_*$  is a strict, isolated minimizer of  $\phi(x, \sigma)$ . We shall also assume, for the time being, that

**AA.11** the sequence  $\{\Delta_k\}$  generated by Algorithm 11.1.1 (or its nonmonotone counterpart) applied to the minimization of (15.3.3) with steps chosen to satisfy AA.1c is such that

$$\Delta = \liminf_{k \rightarrow \infty} \Delta_k > 0.$$

It is then clear that if AA.11 holds, the trust-region constraint will be inactive for all sufficiently large  $k$ . In this case, we may disregard the trust-region constraint in (15.3.5) without affecting the solution. Moreover, since  $s = 0$  is a local solution to the problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H(x_*, y_*) s \rangle + \langle g(x_*), s \rangle \quad (15.3.9a)$$

$$\text{subject to} \quad A_{\mathcal{E}}(x_*) s + c_{\mathcal{E}}(x_*) = 0, \quad A_{\mathcal{T}}(x_*) s + c_{\mathcal{T}}(x_*) \geq 0, \quad (15.3.9b)$$

Theorem 3.2.7 (p. 42) shows that AO.1b, AO.3, and AO.4 hold for the solution  $s$  of (15.3.6) and that the associated Lagrange multiplier  $y$  satisfies  $\sigma > \|y\|_{\mathbb{D}}$  for all  $k$  sufficiently large. But then, provided that we assume that

**AA.1d** the step  $s_k$  is ultimately chosen as the locally unique minimizer of the model (15.3.5) for all  $x_k - x_*$  sufficiently small,

Theorem 15.3.2 implies that  $s_k$  is also a strict, isolated local minimizer of (15.3.6) for all large  $k$ , and thus the iteration is equivalent to the SQP method analysed in Section 15.2.2. Hence, under the stated assumptions, in view of Theorem 15.2.2 we would expect that, given suitable Lagrange multiplier estimates, the method will converge Q-superlinearly. Notice that there is nothing in theory to prevent a step satisfying AA.1c from jumping to a neighbourhood of another local minimizer and thus violating AA.1d, although this is unlikely to happen in practice. In particular, as a consequence of Theorem 15.3.2, the model (15.3.5) has a locally unique solution, and one would hope that it is this solution that is chosen.<sup>241</sup> Thus AA.1d is probably a realistic assumption. But is it reasonable to assume that AA.11 holds?

One aspect we have not considered so far is the effect that the merit function may have on the rate of convergence. For unconstrained minimization, where the objective function is the most obvious merit function, we saw in Theorem 6.5.5 (p. 146) that, under suitable, realistic assumptions, the merit function does not interfere with the asymptotic rate of convergence of the underlying locally convergent iteration. Thus, for instance, if the model is a second-order Taylor approximation, the underlying iteration is an approximation to Newton's method and a Q-superlinear or even Q-quadratic rate of convergence is normally attainable. In our haste to place SQP methods in a globally convergent framework, we have not yet paused to see what the consequences of the globalization are. Specifically, can we be sure that, under ideal but realistic circumstances, our globalized SQP method will converge at a Q-superlinear rate? The answer is, bluntly, no.

For consider the problem

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \quad f(x) = 2(x_1^2 + x_2^2 - 1) - x_1 \quad \text{subject to} \quad c(x) = x_1^2 + x_2^2 - 1 = 0, \quad (15.3.10)$$

---

<sup>241</sup>This depends on the quadratic programming method used to solve (15.3.5). With active set methods, in which monotonic descent of the objective function is ensured, this will happen as soon as the starting point is sufficiently close to the solution. For many interior-point methods, the iterates may initially wander away from such a point.

whose optimal solution is  $(1, 0)^T$  with the optimal Lagrange multiplier  $y_* = \frac{3}{2}$ . It is easy to show that assumptions AO.1b, AO.3, and AO.4 hold at this solution and that the solution to the SQP subproblem (15.2.5), starting from the feasible point  $x_k = (\cos \theta, \sin \theta)^T$  and using the optimal Lagrange multiplier  $y_k = y_*$ , for which  $H_k = I$ , is  $s_k = (\sin^2 \theta, -\cos \theta \sin \theta)^T$ . But in this case  $c(x_k + s_k) = \sin^2 \theta$  and  $f(x_k + s_k) - f(x_k) = \sin^2 \theta$ . Hence

$$\Phi(x_k + s_k) > \Phi(x_k)$$

for any such  $x_k \neq x_*$  and any penalty function—differentiable or otherwise—of the form (14.6.1) (p. 616). Thus any method that requires monotonic descent at each iteration may not permit the step  $s_k$  from  $x_k$  regardless of how close  $x_k$  is to  $x_*$ , and indeed may deny the fast asymptotic rate of convergence we might expect from an unfettered SQP method. This serious defect of (14.6.1) was first reported by Maratos (1978) and has since become commonly known as the *Maratos effect*. We illustrate the Maratos effect in Figure 15.3.1.

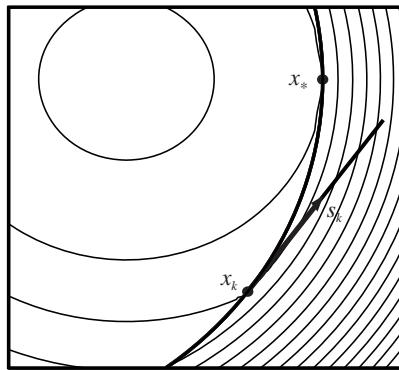


Figure 15.3.1: The Maratos effect for the problem (15.3.10) showing the contours for the  $\ell_1$  penalty function ( $\sigma = 1$ ). Notice how the merit function increases at the point  $x_k + s_k$ .

In the case of Algorithm 11.1.1 (p. 413) applied to the minimization of (15.3.3) with steps chosen to satisfy AA.1c, the Maratos effect may result in  $\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma) < 0$  and thus  $\rho_k < 0$ . The trust-region radius will therefore be reduced to exclude the current model minimizer, and thus  $\Delta_{k+1} \leq \|s_k\|$ . Since  $\|s_k\|$  converges to zero, we see immediately that assumption AA.11 will not, in general, be valid. We now consider ways in which we can overcome this serious disadvantage of the basic method.

### 15.3.2.3 Second-Order Corrections

We have argued that, as far as the asymptotics go, the general problem (15.1.1) might as well be considered to be the equality problem (15.1.2) (we shall write the constraints

as  $c(x) = 0$ ) and that the subproblem (15.3.5) will eventually be equivalent to the quadratic program

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H(x, y)s \rangle + \langle g(x), s \rangle \quad (15.3.11a)$$

$$\text{subject to} \quad A(x)s + c(x) = 0 \quad (15.3.11b)$$

$$\text{and} \quad \|s\| \leq \Delta, \quad (15.3.11c)$$

provided that assumptions AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$ . We shall continue to investigate the case when the trust-region constraint (15.3.11c) is inactive, and in this case the required SQP step satisfies

$$\begin{pmatrix} H(x, y) & A^T(x) \\ A(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ -(y + s^y) \end{pmatrix} = - \begin{pmatrix} g(x) \\ c(x) \end{pmatrix}, \quad (15.3.12)$$

where, for brevity, we have suppressed the iteration subscript  $k$ .

The Maratos effect arises when the curvature of the constraints is not adequately represented by the linearized model (15.2.4b)—a unit step is too large for  $c(x) + A(x)s$  to be an accurate approximation of  $c(x + s)$ . In particular, a Taylor approximation of  $c$  and (15.3.9b) gives that

$$c(x + s) = c(x) + A(x)s + O(\|s\|^2) = O(\|s\|^2), \quad (15.3.13)$$

where the second-order error term  $O(\|s\|^2)$  arises from the constraint curvature. This then suggests that a way around the problem might be to correct for this unforeseen curvature by adding a correction  $s^{\text{CS}}$  to  $x + s$  so that

$$c(x + s + s^{\text{CS}}) = o(\|s\|^2). \quad (15.3.14)$$

But it is equally important that such a correction does not substantially alter  $x + s$ , since, as we have seen, the original step has the potential for fast convergence. Thus we also require that

$$s^{\text{CS}} = o(\|s\|). \quad (15.3.15)$$

A step  $s^{\text{CS}}$  that satisfies (15.3.14) and (15.3.15) is known as a *second-order correction*.

Possibly the simplest second-order correction is obtained by considering the Taylor approximation

$$c(x + s + s^{\text{CS}}) = c(x + s) + A(x + s)s^{\text{CS}} + O(\|s^{\text{CS}}\|^2). \quad (15.3.16)$$

If we choose  $s^{\text{CS}}$  so that

$$c(x + s) + A(x + s)s^{\text{CS}} = 0, \quad (15.3.17)$$

the identity (15.3.16) shows that

$$c(x + s + s^{\text{CS}}) = O(\|s^{\text{CS}}\|^2). \quad (15.3.18)$$

Of course, if  $A(x + s)$  is not full rank, the system may be inconsistent; otherwise, there are many possible solutions to (15.3.17), and we may prefer to find the smallest in some norm. For example, the minimum  $\ell_2$ -norm solution satisfies

$$\begin{pmatrix} I & A^T(x + s) \\ A(x + s) & 0 \end{pmatrix} \begin{pmatrix} s^{cs} \\ -y^{cs} \end{pmatrix} = -\begin{pmatrix} 0 \\ c(x + s) \end{pmatrix} \quad (15.3.19)$$

for some Lagrange multipliers  $y^{cs}$  (see Section 4.4.2). So long as AO.1b holds,  $A(x + s)$  will be of full rank for all  $x$  sufficiently close to  $x_*$ , in which case (15.3.13) and (15.3.19) together imply that

$$s^{cs} = O(\|s\|^2). \quad (15.3.20)$$

Thus, since (15.3.18) and (15.3.20) imply (15.3.14) and (15.3.15), such a solution is suitable as a second-order correction.

There are a large number of variations on this theme. Firstly, we may replace the  $\ell_2$ -norm problem by an appropriate  $H$  norm, in which case (15.3.19) becomes

$$\begin{pmatrix} H & A^T(x + s) \\ A(x + s) & 0 \end{pmatrix} \begin{pmatrix} s^{cs} \\ -y^{cs} \end{pmatrix} = -\begin{pmatrix} 0 \\ c(x + s) \end{pmatrix} \quad (15.3.21)$$

for some other  $y^{cs}$ . (We could even replace the problem with an  $H$  seminorm for which  $N^T(x)HN(x)$  is positive definite, where  $N(x)$  is a basis matrix for the null-space of  $A(x)$ ; again, see Section 4.4.2.) In particular, if  $H = H(x + s, y + s^y)$ , assumptions AO.1b, AO.3, and AO.4 and (15.3.13) and (15.3.21) together imply that (15.3.20) holds, and thus, as before,  $s^{cs}$  is a second-order correction. More generally, we can replace (15.3.21) by

$$\begin{pmatrix} H & A^T(x + s) \\ A(x + s) & 0 \end{pmatrix} \begin{pmatrix} s^{cs} \\ -y^{cs} \end{pmatrix} = -\begin{pmatrix} g \\ c(x + s) \end{pmatrix}$$

and draw the same conclusions so long as  $g = o(\|s\|)$ . One very important case occurs when  $g = g(x + s) - A^T(x + s)(y + s^y)$ —Taylor's approximation gives that

$$g(x + s) - A^T(x + s)(y + s^y) = O(\|s\| \max[\|s\|, \|s^y\|])$$

because  $(s, s^y)$  solves (15.3.12). In this case,

$$\begin{pmatrix} H & A^T(x + s) \\ A(x + s) & 0 \end{pmatrix} \begin{pmatrix} s^{cs} \\ -y^{cs} \end{pmatrix} = -\begin{pmatrix} g(x + s) - A^T(x + s)(y + s^y) \\ c(x + s) \end{pmatrix} \quad (15.3.22)$$

are the criticality conditions for the SQP problem at  $(x + s, y + s^y)$  if  $H = H(x + s, y + s^y)$ , and thus two consecutive SQP steps provide a second-order correction. Globally, of course, we do not expect the solutions of (15.3.5) and (15.3.11) to coincide, and thus we would actually compute  $s^{cs}$  by solving the problem

$$\underset{s^{cs} \in \mathbb{R}^n}{\text{minimize}} \quad m(x + s, H, s^{cs}) \quad (15.3.23)$$

within an appropriate trust region, where  $H$  is a suitable approximation to  $H(x+s, y+s^y)$  and  $m$  is the model (15.3.4).

All of the proposals so far require that the problem functions and their derivatives are reevaluated at  $x+s$ . But this is not necessary. To see this, we replace (15.3.16) by the relationship

$$c(x+s+s^{\text{CS}}) = c(x+s) + A(x)s^{\text{CS}} + O(\|s^{\text{CS}}\| \max[\|s\|, \|s^{\text{CS}}\|])$$

and then aim to choose  $s^{\text{CS}}$  so that

$$c(x+s) + A(x)s^{\text{CS}} = 0 \quad (15.3.24)$$

and hence it follows that

$$c(x+s+s^{\text{CS}}) = O(\|s^{\text{CS}}\| \max[\|s\|, \|s^{\text{CS}}\|]). \quad (15.3.25)$$

The minimum  $\ell_2$ -norm solution to (15.3.24) will, as before, satisfy (15.3.20), in which case (15.3.25) is

$$c(x+s+s^{\text{CS}}) = O(\|s\|^3),$$

and hence this  $s^{\text{CS}}$  is also a second-order correction. The same is true if an  $H$ -seminorm solution is computed, in which case  $s^{\text{CS}}$  satisfies

$$\begin{pmatrix} H & A^T(x) \\ A(x) & 0 \end{pmatrix} \begin{pmatrix} s^{\text{CS}} \\ -y^{\text{CS}} \end{pmatrix} = - \begin{pmatrix} 0 \\ c(x+s) \end{pmatrix}. \quad (15.3.26)$$

A popular variant on this when  $H = H(x, y)$  is suggested by observing that the first block of (15.3.12) may be expressed as

$$H(x, y)s - A^T(x)(y+s^y) + g(x) = 0,$$

which we may combine with (15.3.26) to obtain

$$\begin{pmatrix} H(x, y) & A^T(x) \\ A(x) & 0 \end{pmatrix} \begin{pmatrix} s+s^{\text{CS}} \\ -(y^{\text{CS}}+y+s^y) \end{pmatrix} = - \begin{pmatrix} g(x) \\ c(x+s) - A(x)s \end{pmatrix}.$$

In all of these cases, the only information that is required at  $x+s$  are the values of the constraints; the remaining data need not change, which has the advantage that useful existing factorizations may often be reused. As before, we do not expect the solutions of (15.3.5) and (15.3.11) to coincide far from  $(x_*, y_*)$ , and we should actually compute  $s^{\text{CS}}$  from the subproblem

$$\underset{s^{\text{CS}} \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s^{\text{CS}}, Hs^{\text{CS}} \rangle + \sigma \| (c(x+s) + A(x)s^{\text{CS}})^{\mathcal{I}^-} \|,$$

or equivalently from

$$\underset{s^{\text{CS}} \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \langle g(x), s+s^{\text{CS}} \rangle + \frac{1}{2} \langle s+s^{\text{CS}}, H(s+s^{\text{CS}}) \rangle + \sigma \| (c(x+s) + A(x)s^{\text{CS}})^{\mathcal{I}^-} \|, \quad (15.3.27)$$

within an appropriate trust region, where  $H$  is a suitable approximation to  $H(x, y)$ . It is worth comparing the correction  $s$  computed from (15.3.5) and the  $s+s^{\text{CS}}$  computed

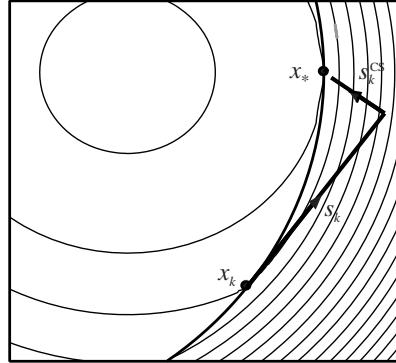


Figure 15.3.2: How the second-order correction (15.3.21) helps avoid the Maratos effect for the problem (15.3.10) with the  $\ell_1$  penalty function. Notice how  $s_k^{CS}$  more than compensates for the increase in the merit function at the point  $x_k + s_k$ , and how much closer  $x_k + s_k + s_k^{CS}$  is to  $x_*$  than  $x_k$  is.

from (15.3.27). The difference in the two subproblems is purely that the constant term for the constraints in the latter is shifted by  $c(x + s) - c(x) - A(x)s$  in an attempt to correct for the curvature of the constraints. We illustrate the beneficial effect of a second-order correction in Figure 15.3.2.

So far, we have defined what we believe to be a useful means of correcting for constraint curvature and have considered a number of ways of computing suitable corrections. It remains for us to show that the addition of a second-order correction actually leads to a useful reduction in the merit function.

We suppose, as before, that  $(x, y)$  is close to  $(x_*, y_*)$  and that AO.1b, AO.3, and AO.4 hold at the latter. Since the step  $s$  is not sufficient to ensure decrease of the merit function  $\phi(x, \sigma)$ , we suppose that we have calculated a second-order correction  $s^{CS}$  and shall investigate the effect of this additional step on the merit function. For convenience, we write

$$\epsilon_x = \|x - x_*\|, \quad \epsilon_y = \|y - y_*\|, \quad \text{and} \quad \epsilon = \max[\epsilon_x, \epsilon_y]$$

and let

$$h(\cdot) = \sigma \|\cdot\|. \quad (15.3.28)$$

We then have the following estimates of the errors that result after the step  $s$ .

**Lemma 15.3.3** Suppose that  $f$  and  $c$  are twice-continuously differentiable and that AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$ . Then the estimates

$$\|x + s - x_*\| = O(\epsilon_x \epsilon), \quad (15.3.29)$$

$$\|s\| = O(\epsilon_x), \quad (15.3.30)$$

$$\|s^y\| = O(\epsilon) \quad (15.3.31)$$

hold for all  $(x, y)$  sufficiently close to  $(x_*, y_*)$ .

**Proof.** Assumptions AO.1b, AO.3, and AO.4 imply that  $K(x, y)$ , defined by (15.2.1), has a uniformly bounded inverse for all  $(x, y)$  sufficiently close to  $(x_*, y_*)$ . In this case, we see, exactly as in the proof of Theorem 15.2.1, that (15.3.29) holds. It then follows immediately that

$$\|s\| \leq \epsilon_x + O(\epsilon_x \epsilon) = O(\epsilon_x),$$

which is (15.3.30). The system (15.3.12) may be written as the equivalent

$$\begin{pmatrix} H(x, y) & A^T(x) \\ A(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ -s^y \end{pmatrix} = - \begin{pmatrix} g(x) - A^T(x)y \\ c(x) \end{pmatrix}. \quad (15.3.32)$$

A Taylor approximation and the criticality of  $(x_*, y_*)$  imply that

$$\begin{pmatrix} g(x) - A^T(x)y \\ c(x) \end{pmatrix} = \begin{pmatrix} g(x_*) - A^T(x_*)y_* + O(\epsilon_x) + O(\epsilon_y) \\ c(x_*) + O(\epsilon_x) \end{pmatrix} = O(\epsilon),$$

which, when combined with (15.3.32) and the bounded inverse of  $K(x, y)$ , leads to (15.3.31).  $\square$

We first show that the largest possible decrease in the merit function that we can expect is proportional to the square of the error in  $x$ .

**Lemma 15.3.4** Suppose that AO.3 and AW.4 hold at  $(x_*, y_*)$ . Then there is a constant  $\kappa_1$  for which

$$\phi(x, \sigma) - \phi(x_*, \sigma) \geq \kappa_1 \epsilon_x^2$$

for all  $x$  in some neighbourhood of  $x_*$ .

**Proof.** Let

$$\kappa_1 = \frac{1}{4} \min_{\substack{\|s\|=1 \\ s \in \mathcal{N}_+}} \langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle$$

and observe that AO.3 implies that  $\kappa_1 > 0$ . Now consider the perturbed problem

$$\begin{aligned} & \text{minimize} && f(x) - \kappa_1 \|x - x_*\|_2^2 \\ & \text{subject to} && c_i(x) = 0 \text{ for } i \in \mathcal{E} \\ & && \text{and} \quad c_i(x) \geq 0 \text{ for } i \in \mathcal{I}, \end{aligned} \quad (15.3.33)$$

and let  $\ell_{\kappa_1}(x, y)$  be its Lagrangian. Since the gradient of  $-\|x - x_*\|_2^2$  vanishes at  $x = x_*$ , while  $\nabla_{xx} \ell_{\kappa_1}(x_*, y_*) = \nabla_{xx} \ell(x_*, y_*) - 2\kappa_1 I$  and  $\langle s, \nabla_{xx} \ell(x_*, y_*) s \rangle \geq 4\kappa_1 > 0$  for all unit vectors  $s \in \mathcal{N}_+$ , it follows immediately that  $(x_*, y_*)$  satisfies the second-order sufficiency conditions AO.3 appropriate for (15.3.33). Thus Theorems 3.2.14 (p. 49) and 14.5.1 (p. 610) show that  $x_*$  is a strict, isolated local minimizer of  $\phi(x, \sigma) - \kappa_1 \|x - x_*\|_2^2$  and thus that

$$\phi(x, \sigma) - \kappa_1 \|x - x_*\|_2^2 \geq \phi(x_*, \sigma)$$

for all  $x$  in some neighbourhood of  $x_*$ , which is the desired result.  $\square$

Next, we show that the maximum decrease in the merit function is accurately predicted by the decrease in the model at the model minimizer.

**Lemma 15.3.5**

$$\phi(x, \sigma) - \phi(x_*, \sigma) = m(x, H, 0) - m(x, H, s) + O(\epsilon_x^2 \epsilon).$$

**Proof.** A Taylor approximation of  $f$  about  $x_*$  yields that

$$f(x_*) = f(x) + \langle g(x), x_* - x \rangle + \frac{1}{2} \langle x_* - x, H(x)(x_* - x) \rangle + O(\epsilon_x^3). \quad (15.3.34)$$

We also have that  $0 = c(x_*) = c(x) + A(x)s$  and thus that  $\partial\|c(x_*)\| = \partial\|c(x) + A(x)s\|$ . In particular, Corollary 11.4.2 (p. 430) gives that

$$\sigma\|c(x_*)\| = -\langle y + s^y, c(x_*) \rangle$$

since  $-(y + s^y)/\sigma \in \partial\|c(x) + A(x)s\|$ . Hence this result and a Taylor approximation of  $c$  about  $x_*$  reveals that

$$\begin{aligned} \sigma\|c(x_*)\| &= -\langle y + s^y, c(x) + A(x)(x_* - x) \rangle \\ &\quad - \frac{1}{2} \left\langle x_* - x, \sum_i (y_i + s_i^y) H_i(x)(x_* - x) \right\rangle + O(\epsilon_x^3) \\ &= -\langle x_* - x - s, A^T(x)(y + s^y) \rangle \quad \text{from (15.3.12)} \\ &\quad - \frac{1}{2} \left\langle x_* - x, \sum_i y_i H_i(x)(x_* - x) \right\rangle + O(\epsilon_x^3) + O(\epsilon_x^2 \|s^y\|) \\ &= -\langle x_* - x - s, A^T(x)(y + s^y) \rangle \\ &\quad - \frac{1}{2} \left\langle x_* - x, \sum_i y_i H_i(x)(x_* - x) \right\rangle + O(\epsilon_x^2 \epsilon), \quad \text{from (15.3.31)} \end{aligned}$$

where  $H_i(x)$  is the Hessian matrix of  $c_i(x)$ . Adding this identity to (15.3.34) and using (15.3.12) and (15.3.29), we find that

$$\begin{aligned} \phi(x_*, \sigma) &= f(x) - \langle x_* - x - s, A^T(x)(y + s^y) \rangle + \langle g(x), x_* - x \rangle \\ &\quad + \frac{1}{2} \langle x_* - x, H(x, y)(x_* - x) \rangle + O(\epsilon_x^2 \epsilon) \\ &= f(x) - \langle g(x) + H(x, y)s, x_* - x - s \rangle + \langle g(x), x_* - x \rangle \\ &\quad + \frac{1}{2} \langle x_* - x, H(x, y)(x_* - x) \rangle + O(\epsilon_x^2 \epsilon) \\ &= f(x) + \langle g(x), s \rangle - \langle x_* - x - s, H(x, y)s \rangle \\ &\quad + \frac{1}{2} \langle x_* - x, H(x, y)(x_* - x) \rangle + O(\epsilon_x^2 \epsilon) \\ &= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \\ &\quad + \frac{1}{2} \langle x_* - x - s, H(x, y)(x_* - x - s) \rangle + O(\epsilon_x^2 \epsilon) \quad (15.3.35) \\ &= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \\ &\quad + O(\|x_* - x - s\|^2) + O(\epsilon_x^2 \epsilon) \\ &= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle + O(\epsilon_x^2 \epsilon). \end{aligned}$$

But

$$m(x, H, 0) - m(x, H, s) = \phi(x, \sigma) - f(x) - \langle g(x), s \rangle - \frac{1}{2} \langle s, H(x, y)s \rangle \quad (15.3.36)$$

since  $c(x) + As = 0$ . Combining (15.3.35) with (15.3.36) gives the required result.  $\square$

Now we show that essentially the same decrease is achieved for the merit function following the second-order correction. Rather than analyse individual second-order corrections, we note that all of the corrections we have discussed may be expressed in the general form

$$\begin{pmatrix} H^{\text{CS}} & A^T(x+p) \\ A(x+p) & 0 \end{pmatrix} \begin{pmatrix} s^{\text{CS}} \\ -y^{\text{CS}} \end{pmatrix} = - \begin{pmatrix} g^{\text{CS}} \\ c(x+s) \end{pmatrix} \quad (15.3.37)$$

for some appropriate  $H^{\text{CS}}$ ,  $p$ , and  $g^{\text{CS}}$ . In order for the resulting  $s$  to be a second-order correction, we require that  $g^{\text{CS}}$  and  $p$  both be small, and in the cases we have considered

$$g^{\text{CS}} = O(\epsilon_x \epsilon) \quad (15.3.38)$$

and

$$p = O(\epsilon_x). \quad (15.3.39)$$

Moreover, for our examples,  $H^{\text{CS}}$  is uniformly positive definite on the null-space of  $A(x+p)$  for all  $(x, y)$  close to  $(x_*, y_*)$ , and this implies that the inverse of

$$K^{\text{CS}} = \begin{pmatrix} H^{\text{CS}} & A^T(x+p) \\ A(x+p) & 0 \end{pmatrix}$$

is uniformly bounded. Globally, of course, we would actually obtain  $s^{\text{CS}}$  by solving the problem

$$\begin{aligned} \underset{s^{\text{CS}} \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) + \langle g^{\text{CS}}, s^{\text{CS}} \rangle + \frac{1}{2} \langle s^{\text{CS}}, H^{\text{CS}} s^{\text{CS}} \rangle \\ & + \sigma \| (c(x+s) + A(x+p)s^{\text{CS}})^{\mathcal{T}-} \| \end{aligned} \quad (15.3.40)$$

within an appropriate trust region. We now consider the ramifications of such a general framework.

**Lemma 15.3.6** Suppose that  $s^{\text{CS}}$  satisfies (15.3.37), where  $g^{\text{CS}}$  and  $p$  satisfy (15.3.38) and (15.3.39), respectively, and that the inverse of  $K^{\text{CS}}$  is uniformly bounded close to  $(x_*, y_*)$ . Then

$$\phi(x, \sigma) - \phi(x + s + s^{\text{CS}}, \sigma) = m(x, H, 0) - m(x, H, s) + O(\epsilon_x^2 \epsilon).$$

**Proof.** First note that, since  $f$  and  $c$  are assumed to be twice-continuously differentiable, the matrices  $A(x)$  and  $H(x)$  are uniformly bounded in the neighbourhood of  $x_*$ . Moreover, it follows from (15.3.13) and (15.3.38) that the right-hand side of (15.3.37) is  $(\epsilon_x \epsilon)$ , and since  $K^{\text{CS}}$  has a bounded inverse, (15.3.37) implies that

$$\|s^{\text{CS}}\| = O(\epsilon_x \epsilon). \quad (15.3.41)$$

A Taylor approximation of  $f$  about  $x + s$ , together with the identities (15.3.30) and (15.3.41), yields that

$$\begin{aligned} f(x + s + s^{\text{CS}}) &= f(x) + \langle g(x), s + s^{\text{CS}} \rangle + \frac{1}{2} \langle s + s^{\text{CS}}, H(x)(s + s^{\text{CS}}) \rangle \\ &\quad + O(\|s + s^{\text{CS}}\|^3) \\ &= f(x) + \langle g(x), s + s^{\text{CS}} \rangle + \frac{1}{2} \langle s, H(x)s \rangle \\ &\quad + O(\epsilon_x^3) + O(\|s\|\|s^{\text{CS}}\|) + O(\|s^{\text{CS}}\|^2) \\ &= f(x) + \langle g(x), s + s^{\text{CS}} \rangle + \frac{1}{2} \langle s, H(x)s \rangle + O(\epsilon_x^2 \epsilon). \end{aligned} \quad (15.3.42)$$

As  $h$  in (15.3.28) is locally Lipschitz, a Taylor approximation of  $c$  about  $x + s$  together with (15.3.30), (15.3.39), and (15.3.41) gives

$$\begin{aligned} |h(c(x + s + s^{\text{CS}})) - h(c(x + s) + A(x + p)s^{\text{CS}})| &\leq \gamma \|c(x + s + s^{\text{CS}}) - c(x + s) - A(x + p)s^{\text{CS}}\| \\ &= \gamma \|(A(x + s) - A(x + p))s^{\text{CS}} + O(\|s^{\text{CS}}\|^2)\| \\ &= O(\|s^{\text{CS}}\|\|s - p\|) + O(\|s^{\text{CS}}\|^2) \\ &= O(\epsilon_x^2 \epsilon) \end{aligned}$$

for some Lipschitz constant  $\gamma$ , and thus

$$h(c(x + s + s^{\text{CS}})) = h(c(x + s) + A(x + p)s^{\text{CS}}) + O(\epsilon_x^2 \epsilon). \quad (15.3.43)$$

Since  $c(x + s) + A(x + p)s^{\text{CS}} = 0 = c(x) + A(x)s$ , it follows that  $\partial\|c(x + s) + A(x + p)s^{\text{CS}}\| = \partial\|c(x) + A(x)s\|$ , and hence Corollary 11.4.2 (p. 430) gives that

$$\begin{aligned} h(c(x + s) + A(x + p)s^{\text{CS}}) &\equiv \sigma \|c(x + s) + A(x + p)s^{\text{CS}}\| \\ &= -\langle y + s^y, c(x + s) + A(x + p)s^{\text{CS}} \rangle, \end{aligned} \quad (15.3.44)$$

since  $-(y + s^y)/\sigma \in \partial\|c(x) + A(x)s\|$ . Thus (15.3.43), (15.3.44), and a Taylor approximation of  $c$  about  $x$  give

$$\begin{aligned} h(c(x + s + s^{\text{CS}})) &= -\langle y + s^y, c(x + s) + A(x + p)s^{\text{CS}} \rangle + O(\epsilon_x^2 \epsilon) \\ &= -\langle y + s^y, c(x) + A(x)(s + s^{\text{CS}}) \rangle \\ &\quad - \frac{1}{2} \left\langle s, \sum_i (y_i + s_i^y) H_i(x)s \right\rangle \\ &\quad + O(\|s\|^3) + O(\|p\|\|s^{\text{CS}}\|) + O(\epsilon_x^2 \epsilon) \\ &= -\langle y + s^y, c(x) + A(x)(s + s^{\text{CS}}) \rangle \quad \text{from (15.3.30), (15.3.39)} \\ &\quad - \frac{1}{2} \left\langle s, \sum_i (y_i + s_i^y) H_i(x)s \right\rangle + O(\epsilon_x^2 \epsilon) \quad \text{and (15.3.41)} \\ &= -\langle A^T(x)(y + s^y), s^{\text{CS}} \rangle \quad \text{from (15.3.12)} \\ &\quad - \frac{1}{2} \left\langle s, \sum_i y_i H_i(x)s \right\rangle + O(\|s^y\|\|s\|^2) + O(\epsilon_x^2 \epsilon) \\ &= -\langle A^T(x)(y + s^y), s^{\text{CS}} \rangle \quad \text{from (15.3.30)} \\ &\quad - \frac{1}{2} \left\langle s, \sum_i y_i H_i(x)s \right\rangle + O(\epsilon_x^2 \epsilon) \quad \text{and (15.3.31)}. \end{aligned}$$

Combining this identity with (15.3.42) we have that

$$\begin{aligned}
\phi(x + s + s^{\text{CS}}, \sigma) &= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \\
&\quad + \langle g(x) - A^T(x)(y + s^y), s^{\text{CS}} \rangle + O(\epsilon_x^2 \epsilon) \\
&= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \\
&\quad + \langle H(x, y)s, s^{\text{CS}} \rangle + O(\epsilon_x^2 \epsilon) \quad \text{from (15.3.12)} \\
&= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \\
&\quad + O(\|s\| \|s^{\text{CS}}\|) + O(\epsilon_x^2 \epsilon) \\
&= f(x) + \langle g(x), s \rangle + \frac{1}{2} \langle s, H(x, y)s \rangle \quad \text{from (15.3.30)} \\
&\quad + O(\epsilon_x^2 \epsilon) \quad \text{and (15.3.41).}
\end{aligned}$$

Combining this with (15.3.36) gives the required result.  $\square$

We then arrive at our central result, namely, that second-order corrections of the forms we have described ensure that the merit function will decrease for all  $(x, y)$  in the neighbourhood of  $(x_*, y_*)$ .

**Theorem 15.3.7** Suppose that assumptions AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$  and that the second-order correction  $s^{\text{CS}}$  and its constituents satisfy (15.3.37), (15.3.38), and (15.3.39). Then for all  $(x, y)$  sufficiently close to  $(x_*, y_*)$ ,

$$\frac{\phi(x, \sigma) - \phi(x + s + s^{\text{CS}}, \sigma)}{m(x, H, 0) - m(x, H, s)} = 1 + O(\epsilon).$$

**Proof.** Lemmas 15.3.4 and 15.3.5 show that<sup>242</sup>

$$m(x, H, 0) - m(x, H, s) \geq \kappa_1 \epsilon_x^2$$

for all  $(x, y)$  sufficiently close to  $(x_*, y_*)$ . The required result then follows from Lemma 15.3.6.  $\square$

As we have already seen, in general we have

$$\frac{\phi(x, \sigma) - \phi(x + s, \sigma)}{m(x, H, 0) - m(x, H, s)} \neq 1 + O(\epsilon)$$

in any neighbourhood of  $(x_*, y_*)$ . Indeed, the Maratos effect implies that this ratio may be negative. Notice that in Theorem 15.3.7 we are comparing the decrease in the model following the initial step with the change in the merit function following the second-order correction. This is slightly different from the normal trust-region framework where model and merit function changes are evaluated at the same point.

We also note that the rate of asymptotic convergence of the basic method is maintained with the second-order correction. This follows from (15.2.7) and (15.3.41) since then

$$\|x + s + s^{\text{CS}} - x_*\| \leq \|x + s - x_*\| + \|s^{\text{CS}}\| = O(\epsilon_x \epsilon). \quad (15.3.45)$$

Of course, extra computation is required. At the very least the constraints need to be reevaluated at  $x + s$ , while for the correction based on (15.3.22), derivatives will

---

<sup>242</sup>Recall the definition of  $O$  in Section 3.3.1.

be required at the intermediate point as well. For this latter case, the convergence may actually be faster than that simply implied by (15.3.45) since the iteration is then equivalent to two consecutive SQP steps.

It remains to show that the second-order correction does not interfere with the global convergence of the method and that the trust region does not ultimately play any role. There are many possible modifications of the basic algorithm that take into account a correction step, but perhaps the most straightforward is Algorithm 11.3.1 (p. 425), which we discussed in Section 11.3.2. Notice that at each iteration of such an algorithm, there is no requirement that a second-order correction be used, or even computed. Indeed, the following strategy suggests itself.

**Algorithm 15.3.1: Possible Step 2 for Algorithm 11.3.1 (p. 425)**

**Step 2: Compute basic step.** Compute a step  $s_k$  satisfying AA.1n for which  $x_k + s_k \in \mathcal{B}_k$  and that reduces the model

$$m(x_k, H_k, s) = f(x_k) + \langle g(x_k), s \rangle + \frac{1}{2} \langle s, H_k s \rangle + \sigma \| (c(x_k) + A(x_k)s)^{\mathcal{T}-} \|.$$

**Step 2a: Compute a correction.** If desired, compute a correction step  $s_k^{\text{CS}}$  that reduces the correction model

$$\begin{aligned} m^{\text{CS}}(x_k, (H_k^{\text{CS}}, s_k, g_k^{\text{CS}}, p_k), s^{\text{CS}}) &= f(x_k) + \langle g_k^{\text{CS}}, s^{\text{CS}} \rangle + \frac{1}{2} \langle s^{\text{CS}}, H_k^{\text{CS}} s^{\text{CS}} \rangle \\ &\quad + \sigma \| (c(x_k + s_k) + A(x_k + p_k)s^{\text{CS}})^{\mathcal{T}-} \| \end{aligned}$$

for appropriate vectors  $g_k^{\text{CS}}$  and  $p_k$  and matrix  $H_k^{\text{CS}}$ , and for which

$$\|s_k + s_k^{\text{CS}}\|_k \leq \tau \Delta_k$$

for some  $\tau > 1$ . If

$$\phi(x_k + s_k + s_k^{\text{CS}}, \sigma) > \phi(x_k + s_k, \sigma)$$

reset  $s_k^{\text{CS}} = 0$ .

**Step 2b: Skip the correction.** Otherwise, set  $s_k^{\text{CS}} = 0$ .

The correction model  $m^{\text{CS}}$  is, as we see from (15.3.40), simply that which we ultimately aim to minimize in order for the second-order correction to provide the asymptotic behaviour predicted by Theorem 15.3.7. As we have suggested, popular choices for the data  $g_k$ ,  $p_k$ , and  $H_k^{\text{CS}}$  are

$$g_k^{\text{CS}} = 0, \quad p_k = 0, \quad \text{and} \quad H_k^{\text{CS}} = H_k \quad \text{or} \quad I \tag{15.3.46}$$

or

$$g_k^{\text{CS}} = g(x_k + s_k), \quad p_k = s_k, \quad \text{and} \quad H_k^{\text{CS}} = H(x_k + s_k, y_k + s_k^y).$$

The choice  $\tau = 2$  is also reasonable.

There are a number of reasons why we might choose to skip Step 2a in favour of 2b. The first is simply that computing a correction involves extra expense. Secondly, in view of this, we may choose not to compute a correction when we are far from the asymptotic stage, although this may be difficult to quantify. In particular, if the basic step lies on the trust-region boundary and the previous corrected step proved to be unsuccessful, this is indicative that we have not arrived in the asymptotics and thus that a correction is not yet justified. Lastly, it may be that the basic step  $s_k$  already gives a reasonable reduction. For example, if

$$\frac{\phi(x_k, \sigma) - \phi(x_k + s_k, \sigma)}{m(x_k, p_k, 0) - m(x_k, p_k, s_k)} \geq \eta_2, \quad (15.3.47)$$

$s_k$  alone provides a very successful step, and there is little reason, or may indeed be little scope, in general, to improve on this.

In order to ensure that the trust-region radius does not interfere with the asymptotics—at least when AO.1b, AO.3, and AO.4 hold—it is clear from Theorem 15.3.7 that so long as a second-order correction is used and the radius is *at some stage* inactive when we are close to  $(x_*, y_*)$ , this behaviour will persist, since  $\rho_k$  converges to 1. Thus all we require is that we have some mechanism for keeping the radius large, at least following successful iterations.

The simplest procedure is to set  $\eta_1 = \eta_2$  and to choose the radius update following very successful iterations to be

$$\Delta_{k+1} = \max[\gamma_3 \Delta_k, \Delta_{\min}] \quad (15.3.48)$$

for some  $0 < \Delta_{\min} \leq \Delta_{\max}$ . For then, by definition, all successful iterations are very successful, and the radius following every very successful iteration is bounded away from zero. Thus, since we know that there are an infinite number of successful iterations—ignoring the special case where only a finite number of iterations are performed—and the SQP corrections  $s_k$  and  $s_k + s_k^{\text{CS}}$  converge to zero, then as soon as  $x_k$  is sufficiently close to  $x_*$  all subsequent corrected SQP iterations will be successful. Notice, however, that we also require that  $y_k$  be close to  $y_*$ , and thus we may need to ensure that this is so at some stage by, for instance, computing  $y_k$  as the least-squares Lagrange multiplier approximation at  $x_k$ ; thereafter, the SQP multiplier estimates will suffice.

Of course, it may not be sensible to pick  $\eta_1 = \eta_2$  throughout, nor to use the above radius update following very successful iterations, and it may be preferable to ensure that these choices are made only when there is some suspicion that we are in the region of fast asymptotic convergence. Other more subtle suggestions have been made. Firstly, if (15.3.47) is not satisfied, then it might be beneficial to compute the correction step to see if the composite step  $s_k + s_k^{\text{CS}}$  proves to be very successful. It may be, however, that evaluating function and derivative values is expensive—in which case a correction using (15.3.46) is likely to be preferred—and, in this case, we may wish to estimate  $\phi(x_k + s_k + s_k^{\text{CS}}, \sigma)$ , using the approximation

$$\begin{aligned} \phi(x_k + s_k + s_k^{\text{CS}}, \sigma) &\approx m^{\text{CS}}(x_k, (H_k^{\text{CS}}, s_k, g_k, t_k), s_k^{\text{CS}}) - m^{\text{CS}}(x_k, (H_k^{\text{CS}}, s_k, g_k, t_k), 0) \\ &\quad + m(x_k, H_k, s_k) \end{aligned}$$

to see if it is actually worth evaluating  $\phi(x_k + s_k + s_k^{\text{cs}}, \sigma)$ . Secondly, if

$$\frac{\phi(x_k, \sigma) - \phi(x_k + s_k, \sigma)}{m(x_k, p_k, 0) - m(x_k, p_k, s_k)} < \eta_1,$$

it is more likely to be worthwhile trying a correction step since  $x_k + s_k$  is definitely not acceptable. In fact, so long as the correction step is skipped when (15.3.47) holds, and is attempted when this test fails, it has been shown that the trust-region radius is ultimately bounded away from zero, and thus fast asymptotic convergence is possible. Since further slight modifications to the basic algorithm are required, and since the proof is rather intricate, we will not give details here.

### Notes and References for Subsection 15.3.2

Early globally convergent SQP methods were based upon the  $\ell_1$  exact penalty function

$$\phi_1(x, \sigma) = f(x) + \sigma \|c(x)^T\|_1 \quad (15.3.49)$$

(see Pschenichny, 1970; Han, 1977; and Powell, 1978). So long as the penalty parameter  $\sigma$  is sufficiently large, the iteration (15.2.8) converges globally with many of the Hessian approximations discussed in the notes at the end of Section 15.2.1. However, despite its simplicity, as we have seen, the iteration has one serious drawback, namely, that the promise of a fast asymptotic rate may be denied by the merit function. This defect was first observed by Maratos (1978). The example (15.3.10) given here is due to Powell (1987). An interesting question is whether the Maratos effect persists. For the example quoted, we have simply seen that there are points arbitrarily close to the solution for which the Maratos effect occurs, but this does not imply that these points will be chosen by a particular algorithm. However, Yuan (1984) gave an example of a min-max problem for which the convergence rate is linear despite it satisfying AO.1b, AO.3, and AO.4, while Fletcher (1985) noted that this example is actually an  $\ell_1$  exact penalty function for a related constrained optimization problem.

Fletcher (1982a, 1987a, Section 14.4) proposed the model (15.2.9) we have considered in this section. Fletcher (1982b) and Yuan (1985a) observed that the Maratos effect may still occur if the search direction is computed by minimizing (15.2.9) but can be prevented if a second-order correction computed from (15.3.27) is made. The idea of using a second-order correction to cope with the Maratos effect first appeared in a number of contemporary papers. Mayne and Polak (1982) proposed adding a second-order correction to the standard SQP direction (15.2.11), Coleman and Conn (1982a, 1982b) preferred using the step (15.2.14) directly, while Fletcher's (1982b) algorithm was designed to cope with the case where the model problem also involves a trust region—it is particularly convenient in this case to choose an  $\ell_\infty$ -norm trust region since the resulting model problem may then still be posed as a quadratic program. A variation on this theme was suggested by Fukushima (1986). All of these techniques allow asymptotic full (but second-order corrected) SQP steps, under a variety of assumptions, and hence encourage Q-superlinear convergence. Fast local convergence properties of such a method were also examined by Womersley (1985) and Yuan (1985b), while Wright (1987, 1989a) showed that it is not necessary to minimize (15.2.9) to full accuracy to achieve fast convergence.

A variety of alternatives to insisting on descent at every iteration for (15.3.49) have been considered. Perhaps the simplest suggestion was that by Chamberlain et al. (1982), in which

the requirement that  $\phi_1(x, \sigma)$  be reduced every iteration is replaced by the requirement that this should happen at least once every  $t > 1$  iterations. Remarkably, this is sufficient to ensure that, so long as a full SQP step is always attempted and provided the iterate is reset to the last “satisfactory” value if more than  $t$  iterations pass without a “satisfactory” reduction, a full SQP step will eventually be “satisfactory” at least every other iteration. This method is known as the “watchdog” technique and may be considered as a variant of the second-order correction (15.3.23). A related proposal by Panier and Tits (1991) was to replace the requirement that  $\phi_1(x, \sigma)$  be sufficiently reduced every iteration by the weaker requirement that the new value be smaller than  $\max\{\phi_1(x, \sigma), \phi_1(x^-, \sigma), \phi_1(x^=, \sigma)\}$ , where  $x^-$  and  $x^=$  are the previous two iterates. They show that such a strategy does not asymptotically prevent full SQP steps. Such nonmonotonic procedures are reminiscent of the algorithm discussed in Section 11.3.1. Similar proposals were given by Zhu (1995), Yamashita and Yabe (1996a), Yang, Zhang, and You (1996), and He, Diao, and Gao (1997).

Related methods based on the  $\ell_\infty$  exact penalty function

$$f(x) + \sigma \|c_I^-(x)\|_\infty$$

have been proposed by Pantoja and Mayne (1991), Heinz and Spellucci (1994), and Yuan (1995).

### 15.3.3 Smooth Exact Penalty Functions

In Section 14.6, we noted that, despite their useful attributes, smooth exact penalty functions like

$$\Phi(x, \sigma) = f(x) - \langle c(x), y(x) \rangle + \frac{1}{2}\sigma \|c(x)\|_2^2, \quad (15.3.50)$$

where

$$y(x) = (A(x)A^T(x))^{-1}A(x)\nabla_x f(x),$$

are likely to be rather cumbersome to manipulate unless one is prepared to work with models that approximate the required derivatives. Of course, we have considered methods that are capable of using approximate derivatives in Section 8.4, and here we briefly consider to what degree the results of that section are applicable to the minimization of (15.3.50). It is straightforward to show, as we noted in the proof of Theorem 14.6.1 (p. 617), that the exact derivatives of (15.3.50) are

$$\nabla_x \Phi(x, \sigma) = \nabla_x \ell(x, y(x, \sigma)) - (\nabla_x y(x))^T c(x)$$

and

$$\begin{aligned} \nabla_{xx} \Phi(x, \sigma) &= \nabla_{xx} \ell(x, y(x, \sigma)) + \sigma A^T(x)A(x) - (\nabla_x y(x))^T A(x) \\ &\quad - A^T(x)(\nabla_x y(x)) - \sum_{i \in \mathcal{E}} (\nabla_{xx} y_i(x)) c_i(x), \end{aligned}$$

where  $y(x, \sigma) = y(x) - \sigma c(x)$  and where Lemma 14.2.1 (p. 577) shows that

$$\nabla_x y(x) = (A(x)A^T(x))^{-1} \left( A(x)\nabla_{xx} \ell(x, y(x)) + E(x) \right), \quad (15.3.51)$$

for which the  $i$ th row of  $E(x)$  is  $(\nabla_x \ell(x, y(x)))^T \nabla_{xx} c_i(x)$ . We now reflect on which, if any, of the terms involved in this gradient and Hessian may usefully be ignored when building an approximation.

Ideally, we hope that the merit function will guide us towards a first-order critical point, in which case both  $c(x)$  and  $\nabla_x \ell(x, y(x))$  (and hence  $R(x)$ ) will approach zero. First consider the gradient,  $\nabla_x \Phi(x, \sigma)$ , and in particular the term  $(\nabla_x y(x))^T c(x)$ . We see from (15.3.51) that this term will likely be dominated near a first-order critical point by

$$\nabla_{xx} \ell(x, y(x)) n^c(x), \text{ where } n^c(x) = A^T(x) (A(x) A^T(x))^{-1} c(x);$$

notice that  $n^c(x)$  gives the minimizer of  $\|A(x)n + c(x)\|_2$  that has minimum norm. Thus, since  $y(x, \sigma) = y(x) - \sigma c(x)$ , an acceptable approximation to the gradient will be

$$\nabla_x \Phi(x, \sigma) \approx \nabla_x \ell(x, y(x)) + \sigma A^T(x) c(x) + \nabla_{xx} \ell(x, y(x)) n^c(x), \quad (15.3.52)$$

or even  $\nabla_x \Phi(x, \sigma) \approx \nabla_x \ell(x, y(x))$  if  $c(x)$  is clearly approaching zero. For the second derivatives, we might neglect all terms involving  $c(x)$  and obtain the approximation

$$\begin{aligned} \nabla_{xx} \Phi(x, \sigma) &\approx \nabla_{xx} \ell(x, y(x)) + \sigma A^T(x) A(x) - (\nabla_x y(x) A(x))^T - A^T(x) (\nabla_x y(x)) \\ &= Q^T(x) \nabla_{xx} \ell(x, y(x)) Q(x) - P^T(x) \nabla_{xx} \ell(x, y(x)) P(x) + \sigma A^T(x) A(x), \end{aligned}$$

where  $P(x)$  and  $Q(x)$  are the orthogonal projection matrices

$$P(x) = A^T(x) (A(x) A^T(x))^{-1} A(x) \text{ and } Q(x) = I - P(x).$$

This approximation to the Hessian has both good and bad features. The terms  $Q^T(x) \nabla_{xx} \ell(x, y(x)) Q(x)$  and  $-P^T(x) \nabla_{xx} \ell(x, y(x)) P(x) + \sigma A^T(x) A(x)$  reflect curvature in two orthogonal spaces (the null-space and range-space of  $A^T(x)$ , respectively). Since the curvature in the first is likely to be positive near a local minimizer in view of second-order necessary conditions (3.2.11) (p. 41), while the same is true for the second term so long as  $\sigma$  is large enough, this decomposition of curvature is easy to interpret. However, neither of the matrices  $Q^T(x) \nabla_{xx} \ell(x, y(x)) Q(x)$  or  $P^T(x) \nabla_{xx} \ell(x, y(x)) P(x)$  likely reflect any sparsity within  $\nabla_{xx} \ell(x, y(x))$  for any but the simplest constraints. This drawback would seem fatal for large-scale problems, but we indicate in the notes at the end of this section that things may not be as bad as at first sight. We pursue this no further.

In practice, we shall shortly see that smooth exact penalty functions are most useful simply as merit functions for some other underlying iteration.

### Notes and References for Subsection 15.3.3

Given the Hessian approximation above, the Newton equations (in the absence of a trust region) are

$$(Q^T(x) H Q(x) - P^T(x) H P(x) + \sigma A^T(x) A(x)) s = -\nabla_x \Phi(x, \sigma), \quad (15.3.53)$$

where  $H$  is a suitable approximation of  $\nabla_{xx} \ell(x, y(x))$ . As  $A(x)$  is of full rank, and letting  $R(x)$  and  $N(x)$  be matrices whose columns span, respectively, the range and null spaces of  $A(x)$ ,

we may write,  $s = R(x)s_r + N(x)s_n$ , just as we did in the notes at the end of Section 15.2.1. Since  $P(x)$  and  $Q(x)$  are orthogonal projectors into the range-space and null-space of  $A^T(x)$ ,

$$\begin{aligned} Q(x)s &= N(x)s_n, & Q(x)N(x) &= N(x), & Q(x)R(x) &= 0, \\ P(x)s &= R(x)s_r, & P(x)N(x) &= 0, & P(x)R(x) &= R(x), \end{aligned}$$

and the Newton equations (15.3.53) may be decomposed into

$$N^T(x)HN(x)s_n = -N^T(x) \left( g(x) + Hn^c(x) \right) \quad (15.3.54)$$

and

$$\begin{aligned} &\left( \sigma R^T(x)A^T(x)A(x)R(x) - R^T(x)HR(x) \right) s_r \\ &= -R^T(x) \left( \nabla_x \ell(x, y) + \sigma A^T(x)c(x) + Hn^c(x) \right), \end{aligned} \quad (15.3.55)$$

where we have used the approximation (15.3.52) to  $\nabla_x \Phi(x, \sigma)$ . Most interestingly, if we compare (15.2.12) and (15.2.13) with (15.3.54), we see that the null-space components  $s_n$  computed from (15.3.54) are identical to those that would be obtained from a standard SQP method. In particular, any presumed impracticality of a method based on  $\Phi$  may not actually arise, since there are good sparsity-exploiting ways of solving the equivalent SQP equations (15.2.11). Turning to (15.3.55) and recalling the definition of  $n^c(x)$ , we then have that

$$\left( \sigma R^T(x)A^T(x)A(x)R(x) - R^T(x)HR(x) \right) (s_r - n_r) = -R^T(x)\nabla_x \ell(x, y).$$

Thus if  $\nabla_x \ell(x, y) = 0$ ,  $s_r = n_r$ , and the complete step  $s$  is the SQP step, while if  $\nabla_x \ell(x, y)$  is nonzero, (15.3.55) defines a correction to the SQP step.

Smooth exact penalty functions were originally proposed as merit functions to globalize iterations based upon the standard SQP direction (15.2.3), using a positive definite approximation  $H_k$ , in a linesearch context. See, for example, Powell and Yuan (1986) and Boggs and Tolle (1989).

## 15.4 Composite-Step Trust-Region SQP Methods

As we have already noted, there are a number of difficulties if we try to impose a trust region on (15.2.4). In particular, the linear constraints (15.2.9b) and the trust region (15.2.9c) may have no common feasible point. The approaches we have considered so far in this chapter have not suffered from this drawback, principally because the step directions used are constructed directly from the merit function and the resulting model problems involve suitably “shifted” linearized constraints. An alternative to these approaches is to try to disaggregate the computation of the part of the step which aims for feasibility from that which aims to reduce a model of the problem. Our intention is not to require that the linearized constraints be satisfied after every step, but more that there be a trend towards feasibility. A very important issue here is ensuring that the model, step, and merit function are all consistent.

All of the methods we shall consider in this section decompose the overall step as

$$s_k = n_k + t_k,$$

where the *normal* step component  $n_k$  provides a move towards feasibility, while the *tangential*<sup>243</sup> step component  $t_k$  reduces the model at the same time as maintaining any gains in feasibility obtained through  $n_k$ . The terms “normal” and “tangential” should be taken descriptively and not literally, since the components may not in fact be orthogonal.<sup>244</sup> We shall call any step that is explicitly formed as the sum of normal and tangential step components a *composite step*.

We shall now explore each of the most important composite-step methods in turn. For simplicity, throughout the majority of this section, we shall only consider the equality-constrained problem (15.1.2), although we shall discuss the general problem in the relevant notes sections, and return to this in general in Section 15.4.4.

### 15.4.1 Vardi-Like Approaches

Perhaps the simplest way around the possible incompatibility of the linearized constraints (15.2.4b) and the trust region (15.2.4c) is to replace the former by relaxed constraints of the form

$$A(x)s + \alpha c(x) = 0, \quad (15.4.1)$$

where  $\alpha \in (0, 1]$  is chosen so that the new constraints and the trust region have a common feasible point. Notice that if  $\alpha = 0$ , the region

$$\mathcal{F}(x, \Delta, \alpha) = \{s \mid A(x)s + \alpha c(x) = 0 \text{ and } \|s\| \leq \Delta\}$$

clearly has a set of feasible points comprising all vectors lying in the null-space of  $A(x)$  whose norm is no greater than  $\Delta$ . The largest suitable value of  $\alpha$  is such that

$$\max_{\alpha \in (0, 1]} \min_{\|s\| \leq \Delta} \|A(x)s + \alpha c(x)\| = 0.$$

If  $A(x)$  is of full rank and an  $\ell_2$ -norm trust region is used, it is easy to show that the largest such  $\alpha$  is

$$\alpha_{\max} = \min \left[ 1, \frac{\Delta}{\|n^c(x)\|_2} \right], \text{ where } n^c(x) \stackrel{\text{def}}{=} -A^T(x)(A(x)A^T(x))^{-1}c(x).$$

The vector  $n^c(x)$  is the solution to  $A(x)n + c(x) = 0$  of a minimum  $\ell_2$  norm, and  $\alpha_{\max}n^c(x)$  is the closest point to the constraints within the trust region; see Section 4.4.2 for methods to compute the projection  $n^c(x)$ . If  $\alpha_{\max} < 1$ , picking  $\alpha = \alpha_{\max}$  will result in  $\mathcal{F}(x, \Delta, \alpha)$  being the single point  $\alpha_{\max}n^c(x)$ .

In practice, it pays to have some “elbow room” in which to allow both normal and tangential components to move. Thus, a value  $\alpha < \alpha_{\max}$  should be chosen if  $\alpha_{\max} < 1$ .

<sup>243</sup>There has been a tendency in the past to refer to these as the vertical and horizontal steps. This unfortunate terminology appears to have originated in Conn and Charalambous (1975), presumably because the authors had a picture in mind in which the two components had the particular orientation alluded to in such epithets. Since this description is hardly invariant to rotation of the coordinate system used, an ad hoc committee formed by Jorge Nocedal in 1997 recommended the terminology that we have adopted here.

<sup>244</sup>For this reason, some of Nocedal’s committee preferred the term *transversal*, while other authors use the term *quasi-normal*, for the normal step.

It is also important not to allow  $\alpha$  to be too much smaller than  $\alpha_{\max}$ , as otherwise there may be little progress towards feasibility. Furthermore, when the problem involves a large number of variables, it is wasteful to compute  $n^C(x)$ , and we might expect to be able to get away with a suitable approximation,  $n^C$ . The normal step, and its consequences are illustrated in Figure 15.4.1.

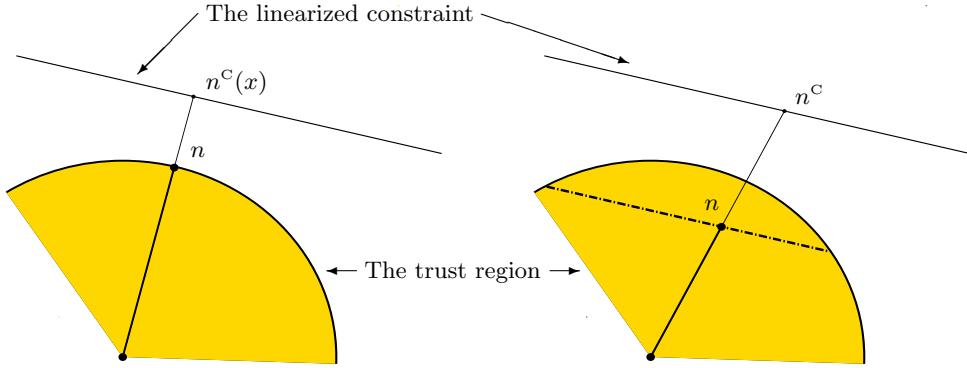


Figure 15.4.1: Computing the normal step. The left-hand side shows  $n^C(x)$  and the largest possible normal step. The right-hand side illustrates a shorter normal step  $n$ , and the freedom this allows for the tangential step—any point on the dotted line is a potential tangential step.

We will shortly derive suitable conditions on both the step  $n^C$  and the allowed values of  $\alpha$ , but first we must consider how to compute the tangential step.

Having found  $n_k = \alpha_k n_k^C$ , the tangential component is chosen so that the linearized constraint infeasibility remains constant, and the flexibility in  $t_k$  is used to reduce a suitable model within the trust region. In particular, the composite step  $s_k$  may be chosen to approximately

$$\begin{aligned} & \underset{s \in \mathbb{R}^n}{\text{minimize}} && \frac{1}{2} \langle s, H_k s \rangle + \langle g(x_k), s \rangle \\ & \text{subject to} && A(x_k)s = A(x_k)n_k \\ & && \|s\| \leq \Delta_k, \end{aligned}$$

or equivalently  $t_k$  may be found to approximately

$$\underset{t \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle t, H_k t \rangle + \langle g(x_k) + H_k n_k, t \rangle \quad (15.4.2a)$$

$$\text{subject to} \quad A(x_k)t = 0 \quad (15.4.2b)$$

$$\text{and} \quad \|n_k + t\| \leq \Delta_k \quad (15.4.2c)$$

for some appropriate approximation,  $H_k$ , to the Hessian of the Lagrangian. This problem always has a feasible point  $s = n_k$ , or equivalently  $t = 0$ . Once again the choice of

the trust-region norm is important. Polyhedral norms give rise to quadratic programming problems. When the  $\ell_2$  norm is used for the trust region in both subproblems, it is sometimes convenient to replace (15.4.2c) by

$$\|t\| \leq \Delta_k^T \stackrel{\text{def}}{=} \sqrt{\Delta_k^2 - \|n_k\|_2^2}. \quad (15.4.3)$$

In this case

$$\|n_k + t\|_2^2 = \|n_k\|_2^2 + \|t\|_2^2 + 2\langle n_k, t \rangle \leq \Delta_k^2 + 2\langle n_k, t \rangle.$$

Hence (15.4.3) ensures that (15.4.2c) is satisfied whenever  $\langle n_k, t \rangle \leq 0$ , while in all cases, a simple calculation reveals that

$$\|n_k + t\|_2 \leq \max_{\|n\|} \|n\| + \sqrt{\Delta_k^2 - \|n\|^2} \leq \sqrt{2}\Delta_k.$$

Thus if one prefers to solve an  $\ell_2$ -norm trust-region problem in which  $t$  lies at the centre, like (15.4.3), rather than off-centred, like (15.4.2c), the composite step is still guaranteed to lie within a fixed fraction of the overall trust region. This is significant since the methods described in Chapter 7 are directly applicable for the centred, but not for the off-centred, problem. Another, slightly more restrictive, possibility is simply to require that

$$\|t\| \leq \Delta_k - \|n_k\|$$

since then it automatically follows that (15.4.2c) holds.

Having set the scene, we shall now build and analyse an algorithm based on these ideas. For simplicity, we shall consider a particular case, namely, where the trust region for both subproblems is specified in the  $\ell_2$  norm. This analysis can easily be extended to other (perhaps weighted) trust-region norms (and merit functions).

#### 15.4.1.1 The Model Problems

As usual, we shall measure progress towards a critical point by ensuring that the generated successful iterates decrease a suitable merit function. Here, a useful merit function is the  $\ell_2$  penalty function

$$\phi(x, \sigma) = f(x) + \sigma\|c(x)\|_2.$$

As we have seen in Section 15.3.2, an overall model of such a function is

$$m(x, H, \sigma, s) = f(x) + \langle g(x), s \rangle + \frac{1}{2}\langle s, Hs \rangle + \sigma\|c(x) + A(x)s\|_2, \quad (15.4.4)$$

but since we aim to compute the composite step in two components, we shall model terms in (15.4.4) separately. With this in mind, we define

$$\begin{aligned} q(x, H, s) &= f(x) + \langle g(x), s \rangle + \frac{1}{2}\langle s, Hs \rangle, \\ m^N(x, n) &= \|c(x) + A(x)n\|_2 \end{aligned}$$

as models of the objective function and constraint infeasibility, as well as

$$m^T(x, H, t) = \langle g(x) + Hn, t \rangle + \frac{1}{2}\langle t, Ht \rangle. \quad (15.4.5)$$

Notice that the decrease in the overall model is

$$\delta m \stackrel{\text{def}}{=} m(x, H, \sigma, 0) - m(x, H, \sigma, s) = \delta m^T + \sigma \delta m^N + \delta q^N, \quad (15.4.6)$$

where

$$\delta m^N = m^N(x, 0) - m^N(x, n)$$

is the decrease in the model  $m^N$  following the normal step,

$$\delta m^T = m^T(x, H, 0) - m^T(x, H, t)$$

is the decrease in the model  $m^T$  following the tangential step, and

$$\delta q^N = q(x, H, 0) - q(x, H, n)$$

is the change in model of the objective function following the normal step.

The first two components on the right-hand side of (15.4.6) will be positive if  $n_k$  is chosen to reduce the linearized infeasibility and  $t_k$  is chosen to reduce its model, but the third term may be positive or negative. Since the normal component is chosen without regard for the model of the objective function, it is to be expected that  $q$  might well increase, although it is (perhaps naively) to be hoped that such an increase will be compensated by a corresponding decrease following the tangential step. As we cannot guarantee that this is so, we may have to resort to increasing  $\sigma$  if we wish to be certain that the overall model  $m(x, H, \sigma, s)$  decreases. For this, we might simply ask that  $\sigma$  be large enough so that

$$\delta m^T + \sigma \delta m^N + \delta q^N \geq 0.$$

Since we prefer to require that a significant decrease occurs, we may ensure this by asking that

$$\delta m = \delta m^T + \sigma \delta m^N + \delta q^N \geq \nu \sigma \delta m^N \quad (15.4.7)$$

for some  $\nu \in (0, 1)$ . In particular, we could compute

$$\sigma^C = -\frac{\delta q^N + \delta m^T}{(1 - \nu) \delta m^N} \quad (15.4.8)$$

as the smallest value for which (15.4.7) is satisfied and replace the current  $\sigma$  by  $\max[\sigma^C, \tau_1 \sigma, \sigma + \tau_2]$  whenever  $\sigma < \sigma^C$  for some appropriate  $\tau_1 \geq 1$  and  $\tau_2 \geq 0$  for which the product  $(\tau_1 - 1)\tau_2 > 0$ . The extra terms  $\tau_1 \sigma$  and  $\sigma + \tau_2$  in the proposed updates are simply to prevent a long sequence of barely increasing penalty parameters. Suitable values might be  $\nu = 0.0001$ ,  $\tau_1 = 2$ , and  $\tau_2 = 1$ . To allow slightly more flexibility, we may prefer to replace  $\sigma^C$  in (15.4.8) by

$$\sigma^C = \max \left[ \sigma^B, -\frac{\delta q^N + \delta m^T}{(1 - \nu) \delta m^N} \right],$$

where  $\sigma^B$  is intended to allow increases in the penalty parameter because of other circumstances. In particular, we know from Theorem 14.5.1 (p. 610) that ultimately we need  $\sigma > \|y_*\|_2$ , where  $y_*$  is the vector of optimal Lagrange multipliers, if we wish

the penalty function to inherit any second-order criticality conditions associated with (15.1.2). We may then wish to pick  $\sigma^B = \|y\|_2$  for some suitable estimates of these optimal values. Of course, if we do not know a suitable value, we may simply set  $\sigma^B = 0$ , and this will not affect the first-order convergence of the method we shall shortly consider. The only crucial property we shall require is that  $\sigma^B$  should remain bounded.

One might be concerned that there may be difficulties here if  $\delta m^N = 0$ . Fortunately, we shall show that this is only possible in our framework if  $n_k = 0$ . As this then implies that  $\delta q^N$  is also zero, we can ensure a decrease in the overall model simply by ensuring that  $t_k$  decreases  $m^T(x, H, t)$ .

#### 15.4.1.2 The Normal and Tangential Steps

We mentioned in the introduction to Section 15.4.1 that we should not expect to compute  $n^C(x)$  exactly, nor to solve the tangential step subproblem very accurately. This is consistent with the approach we have taken throughout this book and is particularly important when the problem involves a large number of unknowns. It is therefore essential that we consider what we mean by an approximate solution to our subproblems. It should not surprise the reader that the requirements we shall impose are based upon, or resemble, the calculation of appropriate Cauchy points.

Consider first the normal step. Recall that our aim here is purely to find a step  $n$  that reduces the infeasibility while at the same time lying within the trust region. We do this by first computing a suitable trial step,  $n_k^C$ , to the linearized constraint, and then, if necessary, scaling  $n_k^C$  so that the resulting step

$$n_k = \alpha_k n_k^C \quad (15.4.9)$$

lies on or within the trust region. Formally, we shall impose the following conditions on the trial step and its scaling.

**AA.1g** There is a constant<sup>245</sup>  $\kappa_{bsc} > 0$  for which the trial step  $n_k^C$  satisfies

$$A(x_k) n_k^C + c(x_k) = 0 \quad \text{and} \quad (15.4.10)$$

$$\|n_k^C\|_2 \leq \kappa_{bsc} \|c(x_k)\|_2 \quad (15.4.11)$$

for all  $k$ .

**AA.1h** The scaling factor  $\alpha_k$  satisfies<sup>246</sup>

$$\alpha_k \in \left[ \min \left[ 1, \frac{\theta \xi^N \Delta_k}{\|n_k^C\|_2} \right], \min \left[ 1, \frac{\xi^N \Delta_k}{\|n_k^C\|_2} \right] \right] \quad (15.4.12)$$

for given parameters  $\xi^N, \theta \in (0, 1]$ .

<sup>245</sup>“bsc” stands for “bound on the step by the constraint”.

<sup>246</sup>The interval in (15.4.12) should be interpreted as the single value  $\alpha_k = 1$  if  $n_k^C = 0$ .

We note that, because of (15.4.9),

$$n_k = 0 \quad \text{if } c(x_k) = 0 \quad (15.4.13)$$

since AA.1g requires that  $n_k^C = 0$ .

The requirements in AA.1g are easy to explain. Firstly, since the step is intended to find a feasible point if at all possible, it should satisfy the linearized constraints. Here, therefore, there is an implicit assumption that the linearized constraints are consistent—this is perhaps the biggest weakness of the whole approach. Secondly, since there is (usually) a vast number of different steps to the linearized constraints, we wish to ensure that the particular step we choose is not too large. Notice that the assumption will be satisfied by the projection step  $n^c(x_k)$  under reasonable assumptions on  $A(x_k)$  and is intended to allow other values that are not too remote from this, perhaps ideal, step.

The scaling implied by AA.1h is also easy to justify. If the trial step is well inside the trust region,  $\alpha_k = 1$  and  $n_k$  will satisfy both the linearized constraint and the trust region. On the other hand, if the trial step is outside the trust region, the scaling ensures that

$$\theta\xi^N\Delta_k \leq \|n_k\|_2 \leq \xi^N\Delta_k, \quad (15.4.14)$$

and thus the step is within a fixed range of the maximum possible. Although the factor  $\xi^N$  has traditionally been picked as 1, a smaller value allows more freedom when computing the tangential step. Reasonable values might be  $\xi^N$  in the range [0.8, 1] and  $\theta = 0.8$ .

When the normal step is so constructed, we have the following result.

**Lemma 15.4.1** Suppose that  $n_k$  is given by (15.4.9), where  $n_k^C$  and  $\alpha_k$  satisfy AA.1g and AA.1h. Then

$$\|c(x_k)\|_2 \geq \delta m_k^N \stackrel{\text{def}}{=} m^N(x_k, 0) - m^N(x_k, n_k) \geq \min \left[ \|c(x_k)\|_2, \frac{\theta\xi^N}{\kappa_{\text{bsc}}} \Delta_k \right] \quad (15.4.15)$$

for all  $k$ .

**Proof.** By the definition of  $m^N$  and of the step (15.4.9) and because of the requirement (15.4.10), we have that

$$\begin{aligned} \delta m_k^N &= \|c(x_k)\|_2 - \|c(x_k) + A(x_k)n_k\|_2 \\ &= \|c(x_k)\|_2 - (1 - \alpha_k)\|c(x_k)\|_2 \\ &= \alpha_k\|c(x_k)\|_2. \end{aligned} \quad (15.4.16)$$

The first inequality in (15.4.15) follows directly since (15.4.12) implies that  $\alpha_k \leq 1$ . It also follows from (15.4.12) and (15.4.16) that

$$\delta m_k^N \geq \min \left[ \|c(x_k)\|_2, \frac{\theta\xi^N\Delta_k\|c(x_k)\|_2}{\|n_k^C\|_2} \right],$$

from which we deduce the remainder of (15.4.15) using (15.4.11).  $\square$

From a practical point of view, finding a trial step  $n_k^C$  that satisfies AA.1g is relatively simple. In particular, as we have suggested, the minimum- $\ell_2$ -norm solution,  $n^C(x_k)$ , to the linearized constraints

$$A(x_k)n + c(x_k) = 0 \quad (15.4.17)$$

gives such a solution. If we can afford to factorize  $A(x_k)A^T(x_k)$ , we may find  $n^C(x)$  directly from  $n(x_k) = A^T(x_k)y_k$ , where  $y_k$  solves

$$A(x_k)A^T(x_k)y_k = -c(x_k) \quad (15.4.18)$$

(see Section 4.4.2). Alternatively, if  $n^F$  is any feasible point for (15.4.17) and if we can compute a basis  $N_k$  for the null-space of  $A(x_k)$ , equations (4.4.11) and (4.4.12) (p. 72) show that  $n^C(x) = n^F + N_k n_k^N$ , where

$$N_k^T N_k s_k^N = -N_k^N n^F. \quad (15.4.19)$$

While factorizing  $N_k^T N_k$  is likely no easier than factorizing  $A(x_k)A^T(x_k)$ , an approximate solution to (15.4.19) may also be useful, since  $n^F + N_k n^N$  satisfies (15.4.17) for any  $n^N$ . In particular, we can aim for an approximate solution of (15.4.19) using the conjugate gradient method (see Chapter 5). Note that we still need to be able to form matrix-vector products with the matrix  $N_k^T N_k$ , but that a basis of the form (4.4.13) (p. 72) is particularly convenient for this—using the decomposition (4.4.13), it is easy to show that an initial feasible point for (15.4.17) is given by

$$n^F = P_k \begin{pmatrix} -(A_k^R)^{-1} (c(x_k) + A_k^N n^A) \\ n^A \end{pmatrix}$$

for any vector  $n^A$ . Approximate solutions  $y$  to (15.4.18) are of little use since the resulting  $n = A^T(x_k)y$  does not satisfy (15.4.17). We shall have to wait until Section 15.4.2 to see how to use approximations to (15.4.18).

We now turn to the tangential step. Since there are computational and analytical advantages in placing the tangential step at the centre of a trust region, we shall suppose that we will compute the tangential step to approximately solve

$$\underset{t \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle t, Ht \rangle + \langle g(x) + Hn, t \rangle \quad (15.4.20a)$$

$$\text{subject to} \quad A(x)t = 0 \quad (15.4.20b)$$

$$\text{and} \quad \|t\| \leq \xi^T \Delta \quad (15.4.20c)$$

for some  $0 < \xi^T \leq 1$ . This implies that the composite step will satisfy

$$\|s\| \leq (\xi^N + \xi^T)\Delta \leq 2\Delta$$

and thus  $s$  will lie within a fixed fraction of the trust region. In order to derive a suitable Cauchy step, it is illuminating to transform (15.4.20) to a more familiar form.

To this end, we let  $N(x)$  be a basis for the null-space of  $A(x)$  (see Section 4.4.2) and use (15.4.20b) to express the tangential step as

$$t = N(x)t^N. \quad (15.4.21)$$

Substituting (15.4.21) into (15.4.20), we then require that  $t^N$  approximately solves

$$\underset{t^N}{\text{minimize}} \quad \frac{1}{2}\langle t^N, H^N(x)t^N \rangle + \langle g^N(x), t^N \rangle \quad (15.4.22a)$$

$$\text{subject to} \quad \|N(x)t^N\| \leq \xi^T \Delta, \quad (15.4.22b)$$

where

$$H^N(x) = N^T(x)HN(x) \quad \text{and} \quad g^N(x) = N^T(x)(g(x) + Hn).$$

This is an unconstrained trust-region problem (in a weighted norm) of the form discussed in Chapter 7. The Cauchy step  $t^C$  for this problem is then simply the minimizer of the model (15.4.5) in the constrained steepest-descent direction  $t^S = -N(x)g^N(x) = -N(x)N^T(x)(g(x) + Hn)$ , within the trust region  $\|t\| \leq \xi^T \Delta$ . That is,

$$t^C = \arg \min_{\alpha \geq 0} \frac{1}{2}\alpha^2 \langle t^S, Ht^S \rangle + \langle g(x) + Hn, \alpha t^S \rangle \quad \text{subject to} \quad \alpha \|t^S\| \leq \xi^T \Delta.$$

Equivalently we can consider a step of the form (15.4.21) with  $t^N = -\alpha g^N(x)$  and where  $\alpha$  solves the problem

$$\underset{\alpha \geq 0}{\text{minimize}} \quad \frac{1}{2}\alpha^2 \langle g^N(x), H^N(x)g^N(x) \rangle - \alpha \|g^N(x)\|_2^2 \quad \text{subject to} \quad \alpha \|N(x)g^N(x)\| \leq \xi^T \Delta.$$

For such a step, we have the following result.

**Lemma 15.4.2** Suppose that  $t^C$  is the Cauchy step for (15.4.22). Then

$$m^T(x, H, 0) - m^T(x, H, t) \geq \frac{1}{2}\|g^N(x)\| \min \left[ \frac{\xi^T \Delta}{\|N(x)\|}, \frac{\|g^N(x)\|}{\beta^T} \right],$$

where  $\beta^T = 1 + \|H^N(x)\|_2$ .

**Proof.** This is a direct application of Corollary 6.3.2 (p. 127) to the problem (15.4.22).  $\square$

In view of Lemma 15.4.2, and since we may expect to decrease the model at least as much as at the Cauchy point, we shall require a reduction on the tangential-step model of the following form.

**AA.1i** For all  $k$ , the tangential step  $t_k$  gives a model reduction

$$\begin{aligned} \delta m_k^T &\stackrel{\text{def}}{=} m^T(x_k, H_k, 0) - m^T(x_k, H_k, t_k) \\ &\geq \kappa_{\text{tmd}} \|g^N(x_k)\| \min \left[ \frac{\xi^T \Delta_k}{\|N(x_k)\|}, \frac{\|g^N(x_k)\|}{\beta_k^T} \right] \end{aligned}$$

for some constant<sup>247</sup>  $\kappa_{\text{tdc}} \in (0, 1)$ , where  $\beta_k^T = 1 + \|H_k^N\|_2$  and  $H_k^N \stackrel{\text{def}}{=} N^T(x_k)H_kN(x_k)$ .

Notice that the Cauchy step actually depends on the basis  $N(x)$  chosen for the null-space of  $A(x)$ . Ideally, an orthonormal basis would be preferred, but the cost of obtaining such a basis may be prohibitive, and one of the other options considered in Section 4.4.2 will be used. All that we shall require is that

**AA.12** there is a constant<sup>248</sup>  $\kappa_{\text{bns}} > 0$  such that for all  $k$ ,

$$0 < \frac{1}{\kappa_{\text{bns}}} \leq \sigma_{\min}[N(x_k)] \leq \sigma_{\max}[N(x_k)] \leq \kappa_{\text{bns}}. \quad (15.4.23)$$

Practically, finding a tangential step that satisfies AA.1i is straightforward. Obvious examples are the relevant Cauchy step and the model minimizer. More importantly, methods like the Steihaug–Toint truncated conjugated gradient method (Section 7.5.1) and the generalized Lanczos trust-region method (Section 7.5.4) applied to the subproblem (15.4.22) produce a suitable step. It is not even necessary to compute the null-space bases explicitly, since there are variants of the truncated conjugated gradient and generalized Lanczos trust-region methods that do this implicitly (see the notes at the end of Sections 7.5.1 and 7.5.4 for details).

#### 15.4.1.3 The Relaxed Linearization SQP Algorithm

We are now in a position to state our complete relaxed linearization SQP algorithm.

##### Algorithm 15.4.1: A relaxed linearization SQP algorithm

**Step 0: Initialization.** An initial point  $x_0$ , an initial trust-region radius  $\Delta_0 > 0$ , and an initial penalty parameter  $\sigma_{-1} > 0$  are given. The constants  $\eta_1, \eta_2, \gamma_1, \gamma_2, \xi^N, \xi^T, \nu, \tau_1$ , and  $\tau_2$  are also given and satisfy the conditions

$$0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1, \quad 0 < \xi^N, \quad \xi^T \leq 1, \quad (15.4.24)$$

$$0 < \nu < 1, \quad \tau_1 \geq 1, \quad \tau_2 \geq 0 \quad \text{and} \quad (\tau_1 - 1)\tau_2 > 0. \quad (15.4.25)$$

Compute  $f(x_0)$  and  $c(x_0)$ , and set  $k = 0$ .

**Step 1: Composite step calculation.** Compute the composite step  $s_k = n_k + t_k$  as follows:

**Step 1a: Normal step calculation.** Compute a step  $n_k = \alpha_k n_k^C$  for which  $\|n_k\| \leq \xi^N \Delta_k$  is satisfied and AA.1g and AA.1h hold.

**Step 1b: Tangential step calculation.** Define a model  $m_k^T$ , and compute a step  $t_k$  for which  $\|t_k\| \leq \xi^T \Delta_k$  is satisfied and AA.1i holds.

<sup>247</sup>“tmd” stands for “tangential-step model decrease”.

<sup>248</sup>“bns” stands for “bounded conditioning of the null-space”.

**Step 2: Increase the penalty parameter if necessary.** Compute the model decreases  $\delta m_k^N$  and  $\delta m_k^T$ , as well as the change in the quadratic model  $\delta q_k^N$  following the normal step. Find

$$\sigma_k^C = \max \left[ \sigma_k^B, -\frac{\delta q_k^N + \delta m_k^T}{(1-\nu)\delta m_k^N} \right]$$

with some suitable  $\sigma_k^B$ , set

$$\sigma_k = \begin{cases} \max[\sigma_k^C, \tau_1 \sigma_{k-1}, \sigma_{k-1} + \tau_2] & \text{if } \sigma_{k-1} < \sigma_k^C, \\ \sigma_{k-1} & \text{otherwise,} \end{cases}$$

and compute the overall model values

$$m(x_k, H_k, \sigma_k, 0) \equiv \phi(x_k, \sigma_k) \text{ and } m(x_k, H_k, \sigma_k, s_k).$$

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$ ,  $c(x_k + s_k)$ , and  $\phi(x_k + s_k, \sigma_k)$ , and define the ratio

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$  and possibly update  $H_{k+1}$ ; otherwise define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (15.4.26)$$

Increment  $k$  by 1 and go to Step 1.

Reasonable values for the parameters in Algorithm 15.4.1 might be

$$\eta_1 = 0.01, \eta_2 = 0.75, \gamma_1 = \gamma_2 = \frac{1}{2}, \xi^N = 1, \xi^T = 0.8, \nu = 0.0001, \tau_1 = 2, \text{ and } \tau_2 = 1, \quad (15.4.27)$$

but other values may be preferred.

#### 15.4.1.4 Global Convergence Analysis

We now consider the global convergence behaviour of Algorithm 15.4.1. Our aim is to show that, under ideal conditions, the iterates converge to a first-order critical point for problem (15.1.2); for our analysis, such a point is most conveniently stated as an  $x_*$  for which

$$c(x_*) = 0 \text{ and } N^T(x_*)g(x_*) = 0 \quad (15.4.28)$$

(see Theorem 3.2.4 [p. 40] and the identity (3.2.10) [p. 41]).

Throughout our analysis, we shall require that the following differentiability and boundedness condition holds.

**AW.1c** The first derivatives of the functions  $f(x)$  and  $c_i(x)$  exist and are Lipschitz continuous in some open set  $\Omega$  containing all the iterates generated by the algorithm.

We shall also require a suitable bound on the second derivative approximations generated by the algorithm. Sometimes, we simply require that these approximations remain bounded, and assume AM.4j. On other occasions we require a suitable bound of the form

$$\|H_k^N\| \leq \kappa_{\text{ubr}} - 1 \quad (15.4.29)$$

(for all  $k$  and some  $\kappa_{\text{ubr}} \geq 1$  independent of  $k$ ) on the *reduced* second derivative approximations  $\{H_k^N\}$ . Such a bound is implied by AM.4j and AA.12 together. For if AM.4j holds,

$$\|H_k\| \leq \kappa_{\text{umh}} \quad (15.4.30)$$

for some  $\kappa_{\text{umh}} \geq 0$ , and hence it follows from (15.4.30) and AA.12 that

$$\|H_k^N\| = \|N^T(x_k)H_kN(x_k)\| \leq \|N(x_k)\|^2\|H_k\| \leq \kappa_{\text{bns}}^2\kappa_{\text{umh}},$$

which is (15.4.29) with  $\kappa_{\text{ubr}} = 1 + \kappa_{\text{bns}}^2\kappa_{\text{umh}}$ . These assumptions also imply that the  $\beta_k^T$  in AA.1i are uniformly bounded by

$$\beta_k^T \leq \kappa_{\text{ubr}}. \quad (15.4.31)$$

As usual, the first step is to investigate the size of the error between the penalty function and its overall model at the new iterate  $x_k + s_k$ .

**Lemma 15.4.3** Suppose that AW.1c and AM.4j hold. Then we have for all  $k$  that the step generated by the algorithm satisfies the bound

$$\left| \phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k) \right| \leq \kappa_0(1 + \sigma_k)\|s_k\|^2 \quad (15.4.32)$$

$$\leq \kappa_{\text{ufm}}(1 + \sigma_k)\Delta_k^2 \quad (15.4.33)$$

for some constants<sup>249</sup>  $\kappa_0$  and  $\kappa_{\text{ufm}} \stackrel{\text{def}}{=} \xi^2\kappa_0$ .

**Proof.** The Lipschitz continuity of  $g(x)$  and Theorem 3.1.4 (p. 29) give that

$$|f(x_k + s_k) - f(x_k) - \langle g(x_k), s_k \rangle| \leq \kappa_{\text{lcg}}\|s_k\|_2^2,$$

for some Lipschitz constant<sup>250</sup>  $\kappa_{\text{lcg}}$ . Similarly, the Lipschitz continuity of  $A(x)$  and Theorem 3.1.6 (p. 29) imply that

$$\|c(x_k + s_k) - A(x_k)s_k - c(x_k)\|_2 \leq \kappa_{\text{lca}}\|s_k\|_2^2$$

<sup>249</sup>“ufm” stands for “upper bound on the difference between function and model”.

<sup>250</sup>“lcg” stands for “Lipschitz constant for the gradient”.

for another Lipschitz constant<sup>251</sup>  $\kappa_{\text{lca}}$ . Combining these inequalities with the bound

$$\langle s_k, H_k s_k \rangle \leq \kappa_{\text{umh}} \|s_k\|_2^2,$$

which follows from AM.4j and (15.4.30), gives (15.4.32) with  $\kappa_0 = \max[\kappa_{\text{lcg}} + \frac{1}{2}\kappa_{\text{umh}}, \kappa_{\text{lca}}]$ . Inequality (15.4.33) then follows directly since

$$\|s_k\|_2 \leq \xi \Delta_k, \quad (15.4.34)$$

where  $\xi = \xi^{\text{N}} + \xi^{\text{T}}$ .  $\square$

The next stage of our proof is, as usual, to deduce that so long as the radius is sufficiently small, a successful step must result from a noncritical point. There is, though, an additional complication here, namely, that there is a possibility that the penalty parameter may continue to increase in Step 2 because of significant increases in the quadratic model  $q(x_k, H_k, n_k)$ . This is reflected in Lemma 15.4.3, where we should only expect better agreement between the penalty function and its model as the radius shrinks if, at the same time, the penalty parameter does not increase too rapidly. We now show that this additional complication does not prevent a successful step from a noncritical point. To do so, we need to prevent the auxiliary penalty parameters  $\{\sigma_k^{\text{B}}\}$  from behaving badly, and we assume that

**AA.13** there is a constant  $\sigma_{\text{max}}^{\text{B}} > 0$  such that

$$0 \leq \sigma_k^{\text{B}} \leq \sigma_{\text{max}}^{\text{B}}$$

for all  $k$ .

**Lemma 15.4.4** Assume that AW.1c, AM.4j, AA.12, and AA.13 hold. Suppose that the iterate  $x_k$  generated by Algorithm 15.4.1 is not a first-order critical point for (15.1.2). Then there exists a  $\Delta_k^0 > 0$  such that  $\rho_k \geq \eta_1$  if  $\Delta_k \in (0, \Delta_k^0)$ .

**Proof.** There are two cases to consider, namely,  $c(x_k) = 0$  and  $c(x_k) \neq 0$ .

Firstly, suppose that  $c(x_k) = 0$ . In this case, (15.4.13), which follows from AA.1g and (15.4.9), requires that  $n_k = 0$  and thus that  $s_k = t_k$ , where  $t_k$  satisfies AA.1i. Since  $\delta m^{\text{N}} = 0 = \delta q^{\text{N}}$  and  $\delta m_k^{\text{T}} > 0$ , (15.4.7) ensures that Step 2 of the algorithm will only increase  $\sigma_k$  on the basis of increases in  $\sigma_k^{\text{B}}$ . If we let  $j_k$  be the index of the last iteration for which  $c(x_{j_k}) \neq 0$ , this observation and AA.13 imply that

$$\sigma_k \leq \sigma_{\text{max}} \stackrel{\text{def}}{=} \max[\tau_1 \max[\sigma_{j_k}, \sigma_{\text{max}}^{\text{B}}], \max[\sigma_{j_k}, \sigma_{\text{max}}^{\text{B}}] + \tau_2].$$

As  $\delta m_k = \delta m_k^{\text{T}}$ , the requirement AA.1i ensures that

$$\delta m_k \geq \kappa_{\text{tmd}} \|g^{\text{N}}(x_k)\| \min \left[ \frac{\xi^{\text{T}} \Delta_k}{\|N(x_k)\|}, \frac{\|g^{\text{N}}(x_k)\|}{\beta_k^{\text{T}}} \right], \quad (15.4.35)$$

<sup>251</sup>“lca” stands for “Lipschitz constant for  $A$ ”.

while the assumption that  $x_k$  is not critical, together with (15.4.28) and  $c(x_k) = 0$ , implies that the reduced gradient  $g^N(x_k) \neq 0$ . Let

$$\Delta_k^0 = \frac{\|g^N(x_k)\|}{\kappa_{\text{bns}}} \min \left[ \frac{1}{\kappa_{\text{ubr}} \xi^T}, \frac{(1 - \eta_1) \kappa_{\text{tmd}} \xi^T}{\kappa_{\text{ufm}} (1 + \sigma_{\max})} \right]. \quad (15.4.36)$$

If  $\Delta_k \leq \Delta_k^0$ , then (15.4.23), (15.4.31), and (15.4.36) imply that

$$\frac{\xi^T \Delta_k}{\|N(x_k)\|} \leq \frac{\|g^N(x_k)\|}{\beta_k^T},$$

and hence (15.4.23) and (15.4.35) give

$$\delta m_k \geq \kappa_{\text{tmd}} \|g^N(x_k)\| \frac{\xi^T \Delta_k}{\|N(x_k)\|} \geq \Delta_k \frac{\kappa_{\text{tmd}} \xi^T \|g^N(x_k)\|}{\kappa_{\text{bns}}}.$$

Combining this with (15.4.33) gives

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \Delta_k \frac{\kappa_{\text{ufm}} \kappa_{\text{bns}} (1 + \sigma_k)}{\kappa_{\text{tmd}} \xi^T \|g^N(x_k)\|}. \quad (15.4.37)$$

But then the fact that  $\Delta_k \leq \Delta_k^0$  as well as (15.4.36) and (15.4.37) imply that  $\rho_k \geq \eta_1$ .

So now suppose instead that  $c(x_k) \neq 0$ . Then Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this and (15.4.15) show that

$$\delta m_k \geq \nu \sigma_k \min \left[ \|c(x_k)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k \right]. \quad (15.4.38)$$

Let

$$\Delta_k^0 = \min \left[ \frac{\kappa_{\text{bsc}} \|c(x_k)\|_2}{\xi^N \theta}, \frac{(1 - \eta_1) \xi^N \theta \nu \sigma_{-1}}{\kappa_{\text{ufm}} \kappa_{\text{bsc}} (1 + \sigma_{-1})} \right]. \quad (15.4.39)$$

If  $\Delta_k \leq \Delta_k^0$ , (15.4.38) and (15.4.39) show that

$$\delta m_k \geq \nu \sigma_k \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k,$$

which combines with (15.4.33) to give

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{\kappa_{\text{ufm}} \kappa_{\text{bsc}} (1 + \sigma_k)}{\nu \xi^N \theta \sigma_k} \Delta_k. \quad (15.4.40)$$

But then the fact that  $\Delta_k \leq \Delta_k^0$  and (15.4.39) and the fact that  $\{\sigma_k\}$  is nondecreasing combines with (15.4.40) to show that once again  $\rho_k \geq \eta_1$ , which proves the lemma.  $\square$

We shall now require the following further assumption.

**AW.1d** The function  $f(x)$  is uniformly bounded from below, the functions  $g(x)$  and  $c(x)$  are uniformly bounded and  $c(x)$  is Lipschitz continuous, at all points computed by the algorithm.

The intention of such an assumption is purely to ensure that the merit function and its derivatives are well behaved at points encountered by the algorithm. In particular, AW.1d implies that

$$f(x_k) \geq f_{\min} \quad (15.4.41)$$

for all  $k$  and some constant  $f_{\min}$ .

Although  $\phi(x, \sigma)$  is a sensible merit function with which to measure convergence, the analysis we shall use is simplified if we define the auxiliary merit function

$$\psi(x, \sigma) = \frac{\phi(x, \sigma) - f_{\min}}{\sigma}.$$

We have the following important result.

**Lemma 15.4.5** Suppose that (15.4.41) holds and that  $\{k_i\}$ ,  $i \geq 0$ , are the indices of the successful iterations resulting from the algorithm. Then

$$\psi(x_{k_i}, \sigma_{k_i}) \geq 0 \quad (15.4.42)$$

and, for any  $j > 0$ ,

$$\psi(x_{k_{i+j}}, \sigma_{k_{i+j}}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \frac{\delta m_{k_i}}{\sigma_{k_i}}. \quad (15.4.43)$$

**Proof.** The first result follows immediately from (15.4.41) and the fact that  $\sigma_k > 0$  for all  $k$ , since then  $\phi(x_k, \sigma_k) \geq f_{\min}$  for any iteration. To prove the second result, since  $k_i$  and  $k_{i+1}$  are consecutive successful iterations, we must have that

$$\phi(x_{k_{i+1}}, \sigma_{k_i}) - f_{\min} = \phi(x_{k_{i+1}}, \sigma_{k_i}) - f_{\min} \leq \phi(x_{k_i}, \sigma_{k_i}) - f_{\min} - \eta_1 \delta m_{k_i}$$

and thus

$$\psi(x_{k_{i+1}}, \sigma_{k_i}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \frac{\delta m_{k_i}}{\sigma_{k_i}}.$$

But the definition of  $\psi$ , the fact that  $\sigma_{k_i} \leq \sigma_{k_{i+1}}$ , and (15.4.41) give that

$$\psi(x_{k_{i+1}}, \sigma_{k_i}) - \psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) = (f(x_{k_{i+1}}) - f_{\min}) \left( \frac{1}{\sigma_{k_i}} - \frac{1}{\sigma_{k_{i+1}}} \right) \geq 0.$$

Combining the last two inequalities gives (15.4.43) for  $j = 1$ , which then implies that  $\psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) \leq \psi(x_{k_i}, \sigma_{k_i})$ . The required result now follows, since then  $\psi(x_{k_{i+j}}, \sigma_{k_{i+j}}) \leq \psi(x_{k_i}, \sigma_{k_i})$ .  $\square$

The first main result we obtain is to show that the algorithm ultimately forces the constraints to zero.

**Theorem 15.4.6** Suppose that AW.1c and AW.1d hold. Then

$$\lim_{k \rightarrow \infty} c(x_k) = 0.$$

**Proof.** Since the result is trivial if  $c(x_\ell) = 0$  for all sufficiently large  $\ell$ , consider an arbitrary infeasible iterate  $x_\ell$ , that is, one for which  $\|c(x_\ell)\|_2 > 0$ . Our first aim is to show that Algorithm 15.4.1 cannot stall at such a point.

The Lipschitz continuity of  $\|c(x)\|_2$  from AW.1d implies that

$$|\|c(x)\|_2 - \|c(x_\ell)\|_2| \leq \gamma \|x - x_\ell\|_2 \quad (15.4.44)$$

for some  $\gamma > 0$  and all  $x$ . Thus (15.4.44) certainly holds for all  $x$  in the open ball

$$\mathcal{O}_\ell \stackrel{\text{def}}{=} \left\{ x \mid \|x - x_\ell\|_2 < \frac{\|c(x_\ell)\|_2}{2\gamma} \right\}. \quad (15.4.45)$$

It then follows immediately from (15.4.44) and (15.4.45) that  $|\|c(x)\|_2 - \|c(x_\ell)\|_2| < \frac{1}{2}\|c(x_\ell)\|_2$  and thus that

$$\|c(x)\|_2 > \frac{1}{2}\|c(x_\ell)\|_2, \quad (15.4.46)$$

which in turn ensures that  $c(x)$  is bounded away from zero for any  $x \in \mathcal{O}_\ell$ . Now, arguing as we did in the second half of Lemma 15.4.4, Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this, (15.4.15), and (15.4.46) show that

$$\delta m_k \geq \nu \sigma_k \min \left[ \|c(x_k)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k \right] \geq \nu \sigma_k \min \left[ \frac{1}{2} \|c(x_\ell)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k \right] \quad (15.4.47)$$

for any  $x_k \in \mathcal{O}_\ell$ . Let

$$\Delta_\ell^0 = \min \left[ \frac{\kappa_{\text{bsc}} \|c(x_\ell)\|_2}{\xi^N \theta}, \frac{(1 - \eta_2) \xi^N \theta \nu \sigma_{-1}}{\kappa_{\text{ufm}} \kappa_{\text{bsc}} (1 + \sigma_{-1})} \right]. \quad (15.4.48)$$

If  $\Delta_k \leq \Delta_\ell^0$ , (15.4.7), (15.4.15), and (15.4.48) show that

$$\delta m_k \geq \nu \sigma_k \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k,$$

which combines with (15.4.33) to give

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{\kappa_{\text{ufm}} \kappa_{\text{bsc}} (1 + \sigma_k)}{\nu \xi^N \theta \sigma_k} \Delta_k. \quad (15.4.49)$$

But then the fact that  $\Delta_k \leq \Delta_\ell^0$ , (15.4.48), and the fact that  $\{\sigma_k\}$  is nondecreasing combine with (15.4.49) to show that  $\rho_k \geq \eta_2$ . Hence, so long as  $\Delta_k \leq \Delta_\ell^0$ , a very successful iteration will occur from any  $x_k \in \mathcal{O}_\ell$ .

Next let  $\mathcal{S}$  be the indices of successful iterates and suppose that all iterates  $x_{k_i}$ ,  $k_i \in \mathcal{S} \geq k_0 \stackrel{\text{def}}{=} \ell$ , remain in  $\mathcal{O}_\ell$ . Then

$$\Delta_{k_i} \geq \Delta^0 \stackrel{\text{def}}{=} \gamma_1 \min[\Delta_\ell, \Delta_\ell^0] \quad (15.4.50)$$

since, as we saw in the previous paragraph, the radius cannot fall below this value without the iteration being very successful—the subsequent radius will be no smaller. Since iteration  $k_i \in \mathcal{S}$  is successful, the bounds (15.4.43), (15.4.47), and (15.4.50) give that

$$\begin{aligned}\psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) &\leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \frac{\delta m_{k_i}}{\sigma_{k_i}} \\ &\leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \nu \min \left[ \frac{1}{2} \|c(x_\ell)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_\ell^0 \right].\end{aligned}$$

Summing over the first  $j$  of these successful iterations, we then have that

$$\psi(x_{k_{i+j}}, \sigma_{k_{i+j}}) \leq \psi(x_{k_i}, \sigma_{k_i}) - j \eta_1 \nu \min \left[ \frac{1}{2} \|c(x_\ell)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_\ell^0 \right].$$

Since the right-hand side of this inequality can be made arbitrarily negative by increasing  $j$ ,  $\psi(x_{k_{i+j}}, \sigma_{k_{i+j}})$  must eventually be negative. However, as this contradicts (15.4.42), our hypothesis that  $x_{k_i}, k_i \in \mathcal{S} \geq \ell$ , remain in  $\mathcal{O}_\ell$  must be false, and thus there must be a first iterate  $x_{k_j}$ ,  $j > 0$ , not in  $\mathcal{O}_\ell$ .

Consider the history of iterates between this  $x_{k_j}$  and  $x_{k_0} = x_\ell$ . Combining (15.4.43) and (15.4.47),

$$\psi(x_{k_k}, \sigma_{k_k}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \nu \min \left[ \frac{1}{2} \|c(x_\ell)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_{k_i} \right] \quad (15.4.51)$$

for any  $i < k \leq j$ , since  $x_{k_i} \in \mathcal{O}_\ell$ . If

$$\Delta_{k_i} \geq \frac{\kappa_{\text{bsc}} \|c(x_\ell)\|_2}{2 \theta \xi^N},$$

(15.4.43) and (15.4.51) with  $k = j$  give that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{1}{2} \eta_1 \nu \|c(x_\ell)\|_2 \leq \psi(x_\ell, \sigma_\ell) - \frac{1}{2} \eta_1 \nu \|c(x_\ell)\|_2. \quad (15.4.52)$$

On the other hand, if

$$\Delta_{k_i} < \frac{\kappa_{\text{bsc}} \|c(x_\ell)\|_2}{2 \theta \xi^N}$$

for all  $0 \leq i \leq j$ ,

$$\psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{\eta_1 \nu \theta \xi^N}{\kappa_{\text{bsc}}} \Delta_{k_i} \quad (15.4.53)$$

for each  $0 \leq i < j$ , and hence, on summing (15.4.53),

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \nu \theta \xi^N}{\kappa_{\text{bsc}}} \sum_{i=0}^{j-1} \Delta_{k_i}. \quad (15.4.54)$$

But as  $x_{k_j} \notin \mathcal{O}_\ell$ , (15.4.34) and (15.4.45) give

$$\frac{\|c(x_\ell)\|_2}{2\gamma} \leq \|x_{k_j} - x_\ell\|_2 \leq \sum_{i=0}^{j-1} \|x_{k_{i+1}} - x_{k_i}\|_2 \leq \xi \sum_{i=0}^{j-1} \Delta_{k_i}, \quad (15.4.55)$$

and therefore (15.4.54) implies that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \nu \theta \xi^N \|c(x_\ell)\|_2}{2\gamma\xi\kappa_{\text{bsc}}}. \quad (15.4.56)$$

Thus in all cases (15.4.52) and (15.4.56) imply that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \kappa_1 \|c(x_\ell)\|_2, \quad \text{where } \kappa_1 \stackrel{\text{def}}{=} \frac{1}{2} \eta_1 \nu \max \left[ 1, \frac{\theta \xi^N}{\gamma \xi \kappa_{\text{bsc}}} \right]. \quad (15.4.57)$$

Since  $\ell$  was arbitrary, and since Lemma 15.4.5 shows that  $\{\psi(x_k, \sigma_k)\}$  is decreasing and bounded from below, (15.4.57) shows that  $\|c(x_\ell)\|_2$  cannot be bounded away from zero.  $\square$

We now turn to the other required first-order criticality condition,  $N^T(x_*)g(x_*) = 0$ . Our first result is that the length of the normal step may be bounded in terms of the overall decrease of the normal model, which itself is bounded by the size of the constraint violation. Since Theorem 15.4.6 showed that the latter converges to zero, so must the former two.

**Lemma 15.4.7** Suppose that AW.1c, AW.1d, AM.4j, and AA.12 hold. Then there is a constant<sup>252</sup>  $\kappa_{\text{bon}} > 0$  such that

$$\kappa_{\text{bon}} \|n_k\|_2 \leq \delta m_k^N \leq \|c(x_k)\|_2 \quad (15.4.58)$$

for all  $k$ .

**Proof.** The result is trivially true if  $c(x_k) = 0$ , since the consequence (15.4.13) of AA.1g and (15.4.9) requires that  $n_k = 0$ . So now assume that  $c(x_k) \neq 0$ . Lemma 15.4.1 gives that

$$\delta m_k^N \geq \min \left[ \|c(x_k)\|_2, \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k \right]. \quad (15.4.59)$$

Now suppose that

$$\Delta_k \leq \frac{\kappa_{\text{bsc}}}{\theta \xi^N} \|c(x_k)\|_2.$$

Then (15.4.59) gives that

$$\delta m_k^N \geq \frac{\theta \xi^N}{\kappa_{\text{bsc}}} \Delta_k \geq \kappa_{\text{bon}} \xi^N \Delta_k \geq \kappa_{\text{bon}} \|n_k\|_2, \quad (15.4.60)$$

since  $\|n_k\|_2 \leq \xi^N \Delta_k$  and where  $\kappa_{\text{bon}} \stackrel{\text{def}}{=} \theta / \kappa_{\text{bsc}}$ . On the other hand, if

$$\Delta_k > \frac{\kappa_{\text{bsc}}}{\theta \xi^N} \|c(x_k)\|_2,$$

---

<sup>252</sup>“bon” stands for “bound on the normal step”.

it follows from (15.4.59) that

$$\delta m_k^N \geq \|c(x_k)\|_2. \quad (15.4.61)$$

However, (15.4.9), the fact that  $\alpha_k \leq 1$  (which is a trivial consequence of (15.4.12)), and (15.4.11) give that

$$\|n_k\| = |\alpha_k| \|n_k^C\| \leq \|n_k^C\| \leq \kappa_{\text{bsc}} \|c(x_k)\|_2.$$

Combining this with (15.4.61) implies

$$\delta m_k^N \geq \frac{1}{\kappa_{\text{bsc}}} \|c(x_k)\|_2 \geq \frac{\theta}{\kappa_{\text{bsc}}} \|c(x_k)\|_2 = \kappa_{\text{bon}} \|n_k\|_2, \quad (15.4.62)$$

since  $\theta \leq 1$ . Thus, as either (15.4.60) or (15.4.62) holds, and since (15.4.15) gives that  $\delta m_k^N \leq \|c(x_k)\|_2$ , the lemma is true.  $\square$

We now show that any fears we may have that the penalty parameter will increase without bound are unfounded.

**Theorem 15.4.8** Suppose that AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold and that (15.4.58) holds for all  $k$  sufficiently large. Then there are constants  $\sigma_{\max} > 0$  and<sup>253</sup>  $\kappa_{\text{bdm}} \geq 0$ , and an index  $k_1$  such that

$$\delta m_k \geq \delta m_k^T + (\sigma_k - \kappa_{\text{bdm}}) \delta m_k^N \quad (15.4.63)$$

and

$$\sigma_k = \sigma_{\max} \quad (15.4.64)$$

for all  $k \geq k_1$ .

**Proof.** In view of (15.4.7) and (15.4.8), Step 2 of Algorithm 15.4.1 ensures that  $\sigma_k$  satisfies

$$\delta m_k = \delta m_k^T + \sigma_k \delta m_k^N + \delta q_k^N \geq \nu \sigma_k \delta m_k^N, \quad (15.4.65)$$

where

$$\delta q_k^N = -\langle g(x_k), n_k \rangle - \frac{1}{2} \langle n_k, H_k n_k \rangle. \quad (15.4.66)$$

Since (i) AW.1d and AM.4j imply that  $g(x_k)$  and  $H_k$  are bounded, (ii) (15.4.58) gives that  $n_k$  is bounded by a multiple of  $\delta m_k^N$ , and (iii) (15.4.58) gives that  $\delta m_k^N \leq \|c(x_k)\|$ , which is itself bounded by AW.1d, we have from (15.4.66) that

$$\delta q_k^N \geq -\kappa_{\text{bdm}} \delta m_k^N$$

for some constant  $\kappa_{\text{bdm}} \geq 0$ . But then (15.4.65) immediately gives (15.4.63), from which we deduce that

$$\delta m_k \geq (\sigma_k - \kappa_{\text{bdm}}) \delta m_k^N$$

---

<sup>253</sup>“bdm” stands for “bound on the decrease in the model of the objective”.

since  $\delta m_k^T > 0$ . Hence (15.4.65) is automatically satisfied for all  $\sigma_k \geq \kappa_{\text{bdm}}/(1 - \nu)$ . Thus, as (15.4.25) and AA.13 hold, every increase of an insufficient  $\sigma_{k-1}$  must be by at least  $\max[\tau_2, (\tau_1 - 1)\sigma_{-1}]$ , and therefore the penalty parameter can only be increased a finite number of times. The required value  $\sigma_{\max}$  is at most the first value of  $\sigma_k$  that exceeds  $\max[\kappa_{\text{bdm}}/(1 - \nu), \sigma_{\max}^B]$ , which proves the lemma.  $\square$

Having seen that the normal step, and consequently the reduction in the normal model, converge to zero, we next investigate the tangential model.

**Lemma 15.4.9** Suppose that AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold and that (15.4.58) holds for all  $k$  sufficiently large. Then there is a constant<sup>254</sup>  $\kappa_{\text{btm}} > 0$  and an index  $k_1$  such that

$$\delta m_k \geq \kappa_{\text{btm}} \delta m_k^T \quad (15.4.67)$$

for all  $k \geq k_1$ .

**Proof.** Consider  $k \geq k_1$  as in Theorem 15.4.8, in which case  $\sigma_k = \sigma_{\max}$ , and let

$$\kappa_{\text{btm}} \stackrel{\text{def}}{=} \frac{1}{2} \min \left[ 1, \frac{\nu \sigma_{\max}}{\kappa_{\text{bdm}} - \sigma_{\max}} \right].$$

Suppose that

$$-\frac{1}{2} \delta m_k^T \leq (\sigma_{\max} - \kappa_{\text{bdm}}) \delta m_k^N.$$

In this case, (15.4.63) gives that  $\delta m_k \geq \frac{1}{2} \delta m_k^T$ , which shows that (15.4.67) is true. Conversely, suppose that

$$-\frac{1}{2} \delta m_k^T > (\sigma_{\max} - \kappa_{\text{bdm}}) \delta m_k^N.$$

Since both  $\delta m_k^T$  and  $\delta m_k^N$  are nonnegative, it must be that  $\sigma_{\max} \leq \kappa_{\text{bdm}}$ , in which case (15.4.7), (15.4.8), and Step 2 of Algorithm 15.4.1 give

$$\delta m_k > \frac{\nu \sigma_{\max}}{\kappa_{\text{bdm}} - \sigma_{\max}} \delta m_k^T.$$

As this is (15.4.67), the proof is complete.  $\square$

At last, we arrive at our second main convergence result.

**Theorem 15.4.10** Suppose that AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold. Then

$$\liminf_{k \rightarrow \infty} \|N^T(x_k)g(x_k)\| = 0.$$

---

<sup>254</sup>“btm” stands for “bound on the tangential model decrease”.

**Proof.** Suppose otherwise, that  $N^T(x_k)g(x_k)$  is bounded away from zero, that is, that

$$\|N^T(x_k)g(x_k)\| \geq 2\kappa_{\text{lbg}} > 0 \quad (15.4.68)$$

for some constant  $\kappa_{\text{lbg}}$  and for all  $k$ . Since  $g^N(x_k) = N^T(x_k)(g(x_k) + H_k n_k)$  and AA.12, AM.4j, (15.4.58), and Theorem 15.4.6 show that  $N^T(x_k)H_k n_k$  converges to zero, (15.4.68) implies that

$$\|g^N(x_k)\| \geq \kappa_{\text{lbg}} > 0 \quad (15.4.69)$$

for all  $k \geq k_2$ . The requirement AA.1i, the inequality (15.4.67), and the bounds (15.4.23) and (15.4.31) ensure that

$$\delta m_k \geq \kappa_{\text{btm}} \kappa_{\text{tmd}} \kappa_{\text{lbg}} \min \left[ \frac{\xi^T \Delta_k}{\kappa_{\text{bns}}}, \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubr}}} \right] \quad (15.4.70)$$

for all  $k \geq k_3 \stackrel{\text{def}}{=} \max[k_1, k_2]$ . It then follows, exactly as in the first part of the proof of Lemma 15.4.4, that if we define

$$\Delta^0 \stackrel{\text{def}}{=} \frac{\kappa_{\text{lbg}}}{\kappa_{\text{bns}}} \min \left[ \frac{1}{\kappa_{\text{ubr}} \xi^T}, \frac{(1 - \eta_2) \kappa_{\text{tmd}} \xi^T}{\kappa_{\text{ufm}} (1 + \sigma_{\text{max}})} \right] \quad (15.4.71)$$

and suppose that  $\Delta_k \leq \Delta^0$ , then (15.4.71) implies that

$$\frac{\xi^T \Delta_k}{\kappa_{\text{bns}}} \leq \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubr}}},$$

and hence (15.4.70) gives

$$\delta m_k \geq \frac{\kappa_{\text{tmd}} \kappa_{\text{lbg}} \xi^T}{\kappa_{\text{bns}}} \Delta_k.$$

Combining this with (15.4.33) gives

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_{\text{max}}) - m(x_k, H_k, \sigma_{\text{max}}, s_k)}{\delta m_k} \right| \leq \Delta_k \frac{\kappa_{\text{ufm}} \kappa_{\text{bns}} (1 + \sigma_{\text{max}})}{\kappa_{\text{tmd}} \xi^T \kappa_{\text{lbg}}}.$$

But then the fact that  $\Delta_k \leq \Delta^0$ , (15.4.71), and (15.4.37) imply that  $\rho_k \geq \eta_2$ . Hence if  $k \geq k_3$  and  $\Delta_k \leq \Delta^0$ , the iteration will be very successful.

We may then prove, exactly as in Theorem 6.4.3 (p. 135), that

$$\Delta_k > \gamma_1 \Delta^0 \quad (15.4.72)$$

for all  $k \geq k_3$ . For suppose otherwise, that iteration  $k$  is the first for which  $\Delta_{k+1} \leq \gamma_1 \Delta^0$ . Then it follows from (15.4.26) that  $\gamma_1 \Delta_k \leq \Delta_{k+1}$ , and hence  $\Delta_k \leq \Delta^0$ . But then, as we concluded at the end of the last paragraph, iteration  $k$  must have been very successful, and thus  $\Delta_k \leq \Delta_{k+1}$ . Since this contradicts the assertion that iteration  $k$  is the first for which  $\Delta_{k+1} \leq \gamma_1 \Delta^0$ , the inequality (15.4.72) must hold.

So now we conclude from (15.4.70) and (15.4.72) that

$$\delta m_k \geq \kappa_3 \stackrel{\text{def}}{=} \kappa_{\text{btm}} \kappa_{\text{tmd}} \kappa_{\text{lbg}} \min \left[ \frac{\xi^T \gamma_1 \Delta^0}{\kappa_{\text{bns}}}, \frac{\kappa_{\text{lbg}}}{\kappa_{\text{ubr}}} \right].$$

Once again, letting  $\mathcal{S}$  denote the indices of successful iterations, we have that

$$\phi(x_k, \sigma_{\max}) - \phi(x_{k+1}, \sigma_{\max}) \geq \eta_1 \delta m_k \geq \eta_1 \kappa_3$$

for all  $k \in \mathcal{S} \geq k_3$ . Summing the above over the set of successful iterations between iteration  $k_3$  and  $k$ , and letting  $\pi_k$  be the number of such successful iterations, we deduce that

$$\phi(x_{k+1}, \sigma_{\max}) \leq \phi(x_1, \sigma_{\max}) - \pi_k \eta_1 \kappa_3.$$

Since AW.1d implies that  $\phi(x, \sigma_{\max})$  is bounded from below, we conclude that  $\pi_k$  is bounded. But this is impossible, since this would imply that there is an index  $\ell$  such that  $x_k = x_\ell$  for all  $k \geq \ell$  after which the trust-region radius must converge to zero, which is not compatible with Lemma 15.4.4. Thus our initial hypothesis (15.4.69) is false, which establishes the theorem.  $\square$

Thus we conclude from Theorems 15.4.6 and 15.4.10 that, under suitable assumptions, if the sequence of iterates generated by Algorithm 15.4.1 has limit points, at least one of them is a first-order critical point. In fact, the result is actually true for all limit points.

**Theorem 15.4.11** Suppose that AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold. Then

$$\lim_{k \rightarrow \infty} \|N^T(x_k)g(x_k)\| = 0.$$

**Proof.** The proof is essentially identical to that of Theorem 6.4.6 (p. 137) but requires the local continuity of  $N(x)$  (see the notes at the end of Section 4.4.2).  $\square$

To summarize, we have seen that in all cases any limit point must be first-order critical (Theorems 15.4.6 and 15.4.11), and moreover, the penalty parameter cannot become arbitrarily large (Theorem 15.4.8).

### Notes and References for Subsection 15.4.1.4

The idea of replacing the linearized constraints by (15.4.1) is due to Vardi (1985) and Byrd, Schnabel, and Shultz (1987). A similar device was proposed by Powell (1978) to handle inconsistent constraints in the basic subproblem (15.2.4). Both Vardi (1985) and Byrd, Schnabel, and Shultz (1987) globalized their algorithms using the  $\ell_1$  exact penalty function, while here we have used the  $\ell_2$  penalty function. Vardi's (1985) analysis was restricted to convex models, while a general theory was given by Byrd, Schnabel, and Shultz (1987). Variations on this theme have also been suggested by Coleman and Yuan (1995), who used the quadratic penalty function and computed the steplength  $\alpha_k$  along the trial normal step  $n_k^C$  by an appropriate linesearch, and by Zhang and Zhu (1990, 1994), Zhang, Zhu, and Fan (1993), and Zhang and Xu (1999b), in which the subproblem (15.4.22) is replaced by the simpler

$$\underset{\|s_n\| \leq \Delta}{\text{minimize}} \quad \frac{1}{2} \langle s_n, H_{nn} s_n \rangle + \langle s_n, N^T \nabla_x \ell(x, y) \rangle.$$

The method of proof we used in this section is influenced by that given by Byrd, Gilbert, and Nocedal (1996) for the Byrd–Omojokun algorithm (see Section 15.4.2), specifically so that large sections need not be reproved when we come to consider this other approach.

#### 15.4.1.5 Fast Convergence

We next turn to the issue of ensuring a reasonable rate of convergence. For simplicity, we shall suppose that the sequence  $\{x_k\}$  has a limit point  $x_*$  (which is a first-order critical point by virtue of Theorem 15.4.11) and that the full-rank assumption AO.1b and the second-order sufficiency condition AO.3 hold at  $x_*$ . We shall consider the case where the iterate  $x_k$  and its corresponding Lagrange multiplier estimate  $y_k$  are close to  $(x_*, y_*)$ .

Let us suppose, for the time being, that the trust-region radius remains inactive in the computation of both normal and tangential steps, that  $H_k = H(x_k, y_k)$ , and that the tangential step is computed accurately, that is, that the subproblem (15.4.20) is solved exactly. The inactivity of the trust region in the normal step computation will ensure that  $n_k = n_k^C$  and thus that

$$A(x_k)n_k + c(x_k) = 0. \quad (15.4.73)$$

Under AO.3, the optimality conditions for (15.4.20) are that

$$H_k t_k + g(x_k) + H_k n_k = A^T(x_k)y_{k+1} \text{ and } A(x_k)t_k = 0$$

when the radius is inactive. Since  $A(x_k)t_k = 0$ , it follows from (15.4.73) that

$$A(x_k)s_k + c(x_k) = 0, \quad (15.4.74)$$

while the remaining condition may be rewritten as

$$H(x_k, y_k)s_k + g(x_k) = A^T y(x_k)y_{k+1}. \quad (15.4.75)$$

But these together imply that  $s_k$  is the standard SQP direction. Since we are using the nonsmooth merit function  $\phi(x, \mu)$ , we know from our discussion in Section 15.3.2.2 that the Maratos effect may disqualify this step and thus deny a fast rate of convergence. The key to avoiding this defect is, as before, to add a suitable second-order correction  $s_k^{CS}$  to  $s_k$  to correct for constraint curvature, and thereby to permit the corrected step  $s_k + s_k^{CS}$ .

We shall only attempt a second-order correction  $s_k^{CS}$  if the original step  $s_k$  did not result in a sufficient decrease in the merit function and if the normal component  $n_k$  is well within the trust region, that is, if  $\rho_k < \eta_1$  and

$$\|n_k\| \leq \chi \xi^N \Delta_k \quad (15.4.76)$$

for some  $\chi \in (0, \theta)$ . The fact that (15.4.14) holds whenever the trial step is outside the trust region, and that this disagrees with (15.4.76), implies that, in this case,  $n_k = n_k^C$  and hence that (15.4.73) and (15.4.74) hold. Thus another way of interpreting

our second condition for the computation of a second-order correction is the entirely reasonable requirement that the original normal step already satisfy the constraints up to first order. The following variant of Algorithm 15.4.1 incorporates a second-order correction.

**Algorithm 15.4.2: Algorithm 15.4.1 with a second-order correction**

The same as Algorithm 15.4.1, except that Step 3 is replaced by

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$ ,  $c(x_k + s_k)$ , and  $\phi(x_k + s_k, \sigma_k)$  and define the ratio

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$  and possibly update  $H_{k+1}$ .

Otherwise, if  $\|n_k\| \leq \chi \xi^N \Delta_k$ ,

**Step 3a: Second-order correction.** Compute any step  $s_k^{CS}$  and redefine the ratio

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k + s_k^{CS}, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k + s_k^{CS}$  and possibly update  $H_{k+1}$ .

Otherwise, define  $x_{k+1} = x_k$ .

Otherwise, define  $x_{k+1} = x_k$ .

Remarkably, the global convergence properties of this modified algorithm do not depend on what second-order correction is used, as we shall now see.

**Theorem 15.4.12** Suppose that Algorithm 15.4.1 is replaced by Algorithm 15.4.2. Then Theorems 15.4.6, 15.4.8, and 15.4.10 continue to hold. If the correction step is also required to satisfy the additional condition

$$\|s_k^{CS}\| \leq \eta^{CS} \Delta_k \tag{15.4.77}$$

for some  $\eta^{CS} > 0$ , Theorem 15.4.11 also continues to hold.

**Proof.** A careful examination of the proofs of Lemmas 15.4.5, 15.4.7, and 15.4.9 and Theorems 15.4.6, 15.4.8, and 15.4.10 shows that the introduction of a second-order correction does not alter the conclusion of these results. The only place we must be careful is in the proof of Theorem 15.4.6, where (15.4.55), and hence (15.4.56), may not hold if one or more second-order corrections are taken since a

correction is not required to lie within the extended trust region. However, a second-order correction can only happen if the original step proves to be unsuccessful. We showed in the second paragraph of the proof of Theorem 15.4.6 that a very successful original step  $s_{k_i}$  must occur from any  $x_{k_i} \in \mathcal{O}$  if  $\Delta_k \leq \Delta_\ell^0$ . Exactly the same argument shows that a successful original step must occur from any such  $x_{k_i}$  if  $\Delta_{k_i} \leq \Delta_\ell^{\min}$ , where

$$\Delta_\ell^{\min} \stackrel{\text{def}}{=} \min \left[ \frac{\kappa_{\text{bsc}} \|c(x_\ell)\|_2}{\xi^N \theta}, \frac{(1 - \eta_1) \xi^N \theta \nu \sigma_{-1}}{\kappa_{\text{ufm}} \kappa_{\text{bsc}} (1 + \sigma_{-1})} \right].$$

Thus, if a second-order correction is required on iteration  $k_i$ , it must be that  $\Delta_{k_i} > \Delta_\ell^{\min}$ . In this case, (15.4.54) implies that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \nu \theta \xi^N}{\kappa_{\text{bsc}}} \Delta_\ell^{\min}$$

since at least one of the  $\Delta_{k_i}$  in the summation is larger than  $\Delta_\ell^{\min}$ . As before, since  $\ell$  was arbitrary and since Lemma 15.4.5 shows that  $\{\psi(x_k, \sigma_k)\}$  is decreasing and bounded from below,  $\Delta_\ell^{\min}$ , and hence  $\|c(x_\ell)\|_2$ , cannot be bounded away from zero. This then establishes Theorem 15.4.6.

In order to deduce that Theorem 15.4.11 still holds, we must once again examine the proof of Theorem 6.4.6 (p. 137). However, we have already reviewed the necessary modifications when proving Theorem 10.4.3 (p. 390) and find that it suffices to require that (15.4.77) holds.  $\square$

We saw in Section 15.3.2.3 that a number of popular second-order corrections, typified by (15.3.37), (15.3.38), and (15.3.39), are sufficient to avoid the Maratos effect and to provide a fast rate of convergence so long as the trust-region radius does not interfere. One way of achieving this was to ensure that the radius after a successful iteration is bounded away from zero, since then eventually a full Newton step will be possible, and the corresponding second-order correction will then guarantee a sufficient decrease in the merit function. Here we shall consider an alternative, which we shall call a partial correction.

Our choice of second-order correction will depend on the fate of the standard SQP step, that is, on the step that satisfies (15.2.3). We say that the standard SQP step  $s_k = n_k + t_k$  is *taken* if the normal component  $n_k$  satisfies the linearized constraints (15.4.73), if  $H^N(x_k)$  is positive definite, and if the tangential component is

$$t_k = N(x_k) t_k^S, \quad \text{where } t_k^S = -(H^N(x_k))^{-1} g^N(x_k) \quad \text{and} \quad \|N(x_k) t_k^S\| \leq \xi^T \Delta_k. \quad (15.4.78)$$

A *full correction*  $s_k^{\text{FC}}$  will be attempted when a second-order correction is warranted and the standard SQP step has been taken. By contrast, a *partial correction*  $s_k^{\text{PC}}$  will be attempted whenever a correction is required but the standard SQP step has not been taken. We aim to show that fast convergence will be possible so long as partial corrections are used when needed. Indeed, as our partial correction is actually a very specific full correction, it will never strictly be necessary to switch to full corrections.

Nevertheless, we may ultimately prefer full corrections when possible since they offer more flexibility.

A full correction will be computed, just as in Section 15.3.2.3, to satisfy

$$\begin{pmatrix} H_k^{\text{CS}} & A^T(x_k + p_k) \\ A(x_k + p_k) & 0 \end{pmatrix} \begin{pmatrix} s_k^{\text{FC}} \\ -y_k^{\text{CS}} \end{pmatrix} = -\begin{pmatrix} g_k^{\text{CS}} \\ c(x_k + s_k) \end{pmatrix} \quad (15.4.79)$$

for some appropriate  $H_k^{\text{CS}}$ ,  $p_k$ , and  $g_k^{\text{CS}}$ . Notice that no trust-region restriction is imposed when computing the correction step, although we shall see later that it may be necessary to discard the step if it is too long. To see what conditions it is reasonable to impose on these quantities, let

$$\epsilon_k^x = \|x_k - x_*\|, \quad \epsilon_k^y = \|y_k - y_*\|, \quad \text{and} \quad \epsilon_k = \max[\epsilon_k^x, \epsilon_k^y].$$

Then if  $s_k$  is the standard SQP step, (15.3.29) gives that

$$\epsilon_k^x \leq \|x_k + s_k - x_*\| + \|s_k\| \leq \|s_k\| + O(\epsilon_k \epsilon_k^x).$$

Hence, if  $(x_k, y_k)$  is sufficiently close to  $(x_*, y_*)$ , we have that  $\epsilon_k^x = O(\|s_k\|)$ , which combines with (15.3.30) to give that (recall the definition of  $\Theta$  in Section 3.3.1)

$$\|s_k\| = \Theta(\epsilon_k^x). \quad (15.4.80)$$

For all the second-order corrections we investigated in Section 15.3.2.3, we saw that (15.3.38) and (15.3.39) hold, and thus, in view of (15.4.80), we shall now require that

$$g_k^{\text{CS}} = O(\epsilon_k \|s_k\|) \quad (15.4.81)$$

and

$$p_k = O(\|s_k\|). \quad (15.4.82)$$

We shall also require that

$$K_k^{\text{CS}} \stackrel{\text{def}}{=} \begin{pmatrix} H_k^{\text{CS}} & A^T(x_k + p_k) \\ A(x_k + p_k) & 0 \end{pmatrix}$$

has a uniformly bounded inverse, which again is consistent with the common choices of  $H_k^{\text{CS}}$  we considered in Section 15.3.2.3. It follows immediately from (15.3.41) that a simple consequence of these requirements is that a full correction satisfies

$$\|s_k^{\text{CS}}\| = O(\epsilon_k \|s_k\|). \quad (15.4.83)$$

We now investigate the effect of such a correction on the merit function.

**Lemma 15.4.13** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.2 have a limit point  $(x_*, y_*)$  at which AO.1b and AO.3 hold. Suppose further that AW.1b and AW.1c hold, that  $H_k = H(x_k, y_k)$  and the standard SQP step  $s_k$  has been taken, that a second-order correction  $s_k^{\text{CS}}$  is required by the algorithm on iteration  $k$ , that  $s_k^{\text{CS}} = s_k^{\text{FC}}$  and its constituents satisfy (15.4.79), (15.4.81), and (15.4.82), respectively, and that the inverse of  $K_k^{\text{CS}}$  is uniformly bounded close to  $(x_*, y_*)$ . Then, for any  $\epsilon > 0$ , there is a radius  $\Delta_{\max} > 0$  such that if  $\|s_k\| \leq \Delta_{\max}$ ,

$$|\phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)| \leq \epsilon \|s_k\|_2^2 \quad (15.4.84)$$

for all  $(x_k, y_k)$  sufficiently close to  $(x_*, y_*)$ .

**Proof.** Theorem 15.4.12 and its consequence, Theorem 15.4.8, show that the penalty parameter  $\sigma_k$  generated by Algorithm 15.4.2 remains bounded. Since the standard SQP step has been taken, the relationship (15.4.84) then follows directly from Lemma 15.3.6 and (15.4.80) by choosing  $(x_k, y_k)$  sufficiently close to  $(x_*, y_*)$ .  $\square$

Now we turn to the partial correction. A partial correction is a very specific full correction, which is intended to correct for constraint curvature without paying further attention to the remaining optimality conditions. This is reasonable, as (15.4.74) shows that  $s_k$  satisfies the linearized constraints whenever a second-order correction is required, while the assumption that  $s_k$  is not a full Newton step violates (15.4.75). We shall choose our partial correction  $s_k^{\text{PC}}$  to satisfy

$$\begin{pmatrix} H_k^{\text{CS}} & A^T(x_k + p_k) \\ A(x_k + p_k) & 0 \end{pmatrix} \begin{pmatrix} s_k^{\text{PC}} \\ -y \end{pmatrix} = - \begin{pmatrix} 0 \\ c(x_k + s_k) \end{pmatrix}, \quad (15.4.85)$$

where  $p_k$  satisfies (15.4.82) as before, but where now the term  $g_k^{\text{CS}}$  used in (15.4.79) has been set to zero. Once again, notice that no trust-region restriction is imposed when computing this step, but that steps that are too long may later be rejected. The most common choice here is  $H_k^{\text{CS}} = I$ , in which case the partial correction is simply the nearest point to the origin on the manifold  $A(x_k + p_k)s + c(x_k + s_k) = 0$  (see Section 4.4.2).

**Lemma 15.4.14** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.2 have a limit point  $(x_*, y_*)$  at which AO.1b and AO.3 hold. Suppose further that AW.1b and AW.1c hold, that  $H_k = H(x_k, y_k)$  but the standard SQP step  $s_k$  has not been taken, that a second-order correction  $s_k^{\text{CS}}$  is required by the algorithm on iteration  $k$ , that  $s_k^{\text{CS}} = s_k^{\text{PC}}$  and its constituents satisfy (15.4.85) and (15.4.82), respectively, and that the inverse of  $K_k^{\text{CS}}$  is uniformly bounded close to  $(x_*, y_*)$ . Then, for any  $\epsilon > 0$ , there is a radius  $\Delta_{\max} > 0$  such that if  $\|s_k\| \leq \Delta_{\max}$ , the bound (15.4.84) holds for all  $(x_k, y_k)$  sufficiently close to  $(x_*, y_*)$ .

**Proof.** It follows by definition and from (15.4.74) that

$$\begin{aligned} & \left| \phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) - m(x_k, H_k, \sigma_k, s_k) \right| \\ &= \left| f(x_k + s_k + s_k^{\text{CS}}) + \sigma_k \|c(x_k + s_k + s_k^{\text{CS}})\|_2 \right. \\ &\quad \left. - f(x_k) - \langle g(x_k), s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle - \sigma_k \|c(x_k) + A(x_k) s_k\|_2 \right| \\ &\leq \left| f(x_k + s_k + s_k^{\text{CS}}) - f(x_k) - \langle g(x_k), s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \right| \\ &\quad + \sigma_k \|c(x_k + s_k + s_k^{\text{CS}})\|_2. \end{aligned} \tag{15.4.86}$$

Since the second-order correction is taken, (15.4.74) and Theorem 3.1.6 (p. 29), using the Lipschitz continuity of  $A(x_k)$  implied by AW.1c, give that

$$\|c(x_k + s_k)\|_2 = O(\|s_k\|^2).$$

The boundedness of the inverse of  $K_k^{\text{CS}}$  and the defining equation (15.4.85) then imply that

$$\|s_k^{\text{CS}}\| = O(\|s_k\|^2). \tag{15.4.87}$$

As  $\|\cdot\|_2$  is locally Lipschitz, a Taylor approximation of  $c$  about  $x_k + s_k$ , together with the second block of (15.4.85), (15.4.82), and (15.4.87), gives

$$\begin{aligned} \|c(x_k + s_k + s_k^{\text{CS}})\|_2 &= \left| \|c(x_k + s_k + s_k^{\text{CS}})\|_2 - \|c(x_k + s_k) + A(x_k + p_k) s_k^{\text{CS}}\|_2 \right| \\ &\leq \gamma \|c(x_k + s_k + s_k^{\text{CS}}) - c(x_k + s_k) - A(x_k + p_k) s_k^{\text{CS}}\|_2 \\ &= \gamma \|(A(x_k + s_k) - A(x_k + p_k)) s_k^{\text{CS}} + O(\|s_k^{\text{CS}}\|^2)\|_2 \\ &= O(\|s_k^{\text{CS}}\| \|s_k - p_k\|) + O(\|s_k^{\text{CS}}\|^2) \\ &= O(\|s_k\|^3) \end{aligned}$$

for some Lipschitz constant  $\gamma$ . Since Theorem 15.4.12 and its consequence, Theorem 15.4.8, again imply that the penalty parameter  $\sigma_k$  generated by Algorithm 15.4.2 is bounded, we deduce that

$$\sigma_k \|c(x_k + s_k + s_k^{\text{CS}})\|_2 = O(\|s_k\|^3). \tag{15.4.88}$$

Turning to the other constituent of (15.4.86), a Taylor expansion of  $f$  about  $x_k + s_k$  reveals that

$$f(x_k + s_k + s_k^{\text{CS}}) = f(x_k + s_k) + \langle g(x_k + s_k), s_k^{\text{CS}} \rangle + \frac{1}{2} \langle s_k^{\text{CS}}, \nabla_{xx} f(x_k + s_k + \theta_1 s_k^{\text{CS}}) \rangle,$$

while another about  $x_k$  gives

$$f(x_k + s_k) = f(x_k) + \langle g(x_k), s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} f(x_k + \theta_2 s_k) \rangle$$

for some constants  $\theta_1$  and  $\theta_2$  in  $[0, 1]$ . Thus these expansions, AW.1b, (15.4.87), and the choice  $s_k^{\text{CS}} = s_k^{\text{PC}}$  give

$$\begin{aligned} & |f(x_k + s_k + s_k^{\text{CS}}) - f(x_k) - \langle g(x_k), s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle| \\ & \leq \frac{1}{2} \left| \langle s_k^{\text{CS}}, \nabla_{xx} f(x_k + s_k + \theta_1 s_k^{\text{CS}}) s_k^{\text{CS}} \rangle \right| \\ & \quad + \frac{1}{2} \left| \langle s_k, (\nabla_{xx} f(x_k + \theta_2 s_k) - \nabla_{xx} f(x_k)) s_k \rangle \right| \\ & \quad + \left| \langle g(x_k + s_k), s_k^{\text{CS}} \rangle + \frac{1}{2} \sum_{i=1}^m [y_k]_i \langle s_k, \nabla_{xx} c_i(x_k) s_k \rangle \right| \\ & = \left| \langle g(x_k + s_k), s_k^{\text{PC}} \rangle + \frac{1}{2} \sum_{i=1}^m [y_k]_i \langle s_k, \nabla_{xx} c_i(x_k) s_k \rangle \right| + O(\|s_k\|^3). \end{aligned} \tag{15.4.89}$$

Now consider the vector  $y_k^{\text{PC}}$  defined by

$$\begin{pmatrix} H_k^{\text{CS}} & A^T(x_k + p_k) \\ A(x_k + p_k) & 0 \end{pmatrix} \begin{pmatrix} s \\ -y_k^{\text{PC}} \end{pmatrix} = - \begin{pmatrix} g(x_k + s_k) \\ 0 \end{pmatrix}. \tag{15.4.90}$$

Note that this equation may be written equivalently as

$$\begin{pmatrix} H_k^{\text{CS}} & A^T(x_k + p_k) \\ A(x_k + p_k) & 0 \end{pmatrix} \begin{pmatrix} s \\ y_* - y_k^{\text{PC}} \end{pmatrix} = - \begin{pmatrix} g(x_k + s_k) - A^T(x_k + p_k)y_* \\ 0 \end{pmatrix}. \tag{15.4.91}$$

Furthermore, Taylor approximations of the Lipschitz continuous  $g$  and  $A$  around  $x_*$ , together with the first-order criticality condition  $g(x_*) - A^T(x_*)y_* = 0$  and the requirement (15.4.82), reveal that

$$\begin{aligned} g(x_k + s_k) - A^T(x_k + p_k)y_* &= g(x_*) - A^T(x_*)y_* \\ &\quad + O(\|x_k + s_k - x_*\|) + O(\|x_k + p_k - x_*\|) \\ &= O(\|x_k - x_*\|) + O(\|s_k\|). \end{aligned}$$

It then follows from (15.4.91) and the boundedness of the inverse of  $K_k^{\text{CS}}$  that

$$\|y_k^{\text{PC}} - y_*\| = O(\|x_k - x_*\|) + O(\|s_k\|).$$

Thus  $y_k^{\text{PC}}$  can be made arbitrarily close to  $y_*$ , and thus to  $y_k$ , by ensuring that  $x_k$  is close to  $x_*$  and  $s_k$  is small, and can therefore be considered to be a useful Lagrange multiplier estimate.

Forming the inner product of both sides of (15.4.85) with  $(s^T - (y_k^{\text{PC}})^T)^T$  and both sides of (15.4.90) with  $((s_k^{\text{PC}})^T - y^T)^T$ , we find that

$$\langle -y_k^{\text{PC}}, c(x_k + s_k) \rangle = \langle s_k^{\text{PC}}, g(x_k + s_k) \rangle. \quad (15.4.92)$$

Moreover, a Taylor approximation of  $c_i$  about  $x_k$  and (15.4.74) give that

$$\begin{aligned} c_i(x_k + s_k) &= c_i(x_k) + \langle \nabla_x c_i(x_k), s_k \rangle + \frac{1}{2} \sum_{i=1}^m \langle s_k, \nabla_{xx} c_i(x_k + \theta_i s_k) s_k \rangle \\ &= \frac{1}{2} \sum_{i=1}^m \langle s_k, \nabla_{xx} c_i(x_k + \theta_i s_k) s_k \rangle \end{aligned} \quad (15.4.93)$$

for some  $\theta_i$  in  $[0, 1]$ . Thus (15.4.92) and (15.4.93) show that

$$\begin{aligned} \langle g(x_k + s_k), s_k^{\text{PC}} \rangle &+ \frac{1}{2} \sum_{i=1}^m [y_k]_i \langle s_k, \nabla_{xx} c_i(x_k) s_k \rangle \\ &= -\langle y_k^{\text{PC}}, c(x_k + s_k) \rangle + \frac{1}{2} \sum_{i=1}^m [y_k]_i \langle s_k, \nabla_{xx} c_i(x_k) s_k \rangle \\ &= \frac{1}{2} \sum_{i=1}^m \left\langle s_k, \left( [y_k]_i \nabla_{xx} c_i(x_k) - [y_k^{\text{PC}}]_i \nabla_{xx} c_i(x_k + \theta_i s_k) \right) s_k \right\rangle \\ &= \frac{1}{2} \sum_{i=1}^m \left\langle s_k, \left( [y_k]_i - [y_k^{\text{PC}}]_i \right) \nabla_{xx} c_i(x_k) s_k \right\rangle \\ &\quad + \left\langle s_k, [y_k^{\text{PC}}]_i \left( \nabla_{xx} c_i(x_k) - \nabla_{xx} c_i(x_k + \theta_i s_k) \right) s_k \right\rangle \\ &= o(\|s_k\|^2) \end{aligned}$$

since  $y_k^{\text{PC}}$  can be made arbitrarily close to  $y_k$  by ensuring that  $x_k$  is close to  $x_*$  and that  $s_k$  is small. Combining this with (15.4.89), we then have that

$$\left| f(x_k + s_k + s_k^{\text{CS}}) - f(x_k) - \langle g(x_k), s_k \rangle - \frac{1}{2} \langle s_k, H_k s_k \rangle \right| = o(\|s_k\|^2),$$

from which we deduce (15.4.84) because of (15.4.86) and (15.4.88).  $\square$

Having studied the crucial properties of the full and partial corrections, we may now state precisely how we intend to embed them within Algorithm 15.4.2.

#### Algorithm 15.4.3: A second-order correction for Algorithm 15.4.2

The same as Algorithm 15.4.2, except that Step 3a is replaced by

**Step 3a: Second-order correction.** Choose  $H_k^{\text{CS}}$  and  $p_k$  so that  $K_k^{\text{CS}}$  is uniformly bounded and (15.4.82) is satisfied.

If  $s_k$  is a standard SQP step, choose  $g^{\text{CS}}$  to satisfy (15.4.81) and compute  $s_k^{\text{CS}} = s_k^{\text{FC}}$  from (15.4.79). Otherwise, compute  $s_k^{\text{CS}} = s_k^{\text{PC}}$  from (15.4.85).

If

$$\|s_k^{\text{CS}}\| \leq \eta^{\text{CS}} \Delta_k, \quad (15.4.94)$$

**Step 3b: Attempt the second-order correction.** Redefine the ratio

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k + s_k^{\text{CS}}$  and possibly update  $H_{k+1}$ .

Otherwise, define  $x_{k+1} = x_k$ .

Otherwise, define  $x_{k+1} = x_k$ .

and Step 4 is replaced by

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \quad x_{k+1} = x_k + s_k, \\ & \text{and } \|n_k\| \leq \chi \xi^N \Delta_k, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{in all other cases when } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (15.4.95)$$

Increment  $k$  by 1 and go to Step 1.

The test (15.4.94) is only needed to ensure that all limit points are first-order critical. Since this is a technical requirement, in practice it would be reasonable to choose a very large  $\eta^{\text{CS}}$ . The slightly more specific Step 4 above is required when the original step is acceptable but the normal component is small, as otherwise the radius may still converge to zero, blocking further progress. Since Algorithm 15.4.3 is just a specific case of Algorithm 15.4.2, the global convergence results given in Theorem 15.4.12 are still valid. We have the following useful asymptotic convergence result.

**Theorem 15.4.15** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.3 have a limit point  $(x_*, y_*)$  at which AO.1b and AO.3 hold. Suppose further that AW.1b, AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold. Finally suppose that  $H_k = H(x_k, y_k)$ , that  $\sigma_k^B = \|y_k\| + \tau$  for some small  $\tau > 0$ , and that the standard SQP step is taken whenever possible for all  $(x_k, y_k)$  close to  $(x_*, y_*)$ . Then  $\{x_k\}$  converges to  $x_*$  Q-superlinearly. If  $\|y_k - y_*\| = O(\|x_k - x_*\|)$ , the rate is Q-quadratic.

**Proof.** The assumptions are sufficient for Theorem 15.4.12 to ensure that  $x_*$  is a first-order critical point and that the penalty parameter  $\sigma_k$  is ultimately fixed at some  $\sigma_{\max} > \|y_*\|_2$ . It then follows from AO.1b and AO.3 and Theorems 14.5.1 (p. 610) and 3.2.14 (p. 49) that  $x_*$  is a strict, isolated local minimizer of  $\phi(x, \sigma_{\max})$ .

Since the sequence  $\{\phi(x_k, \sigma_{\max})\}$  must decrease monotonically for all sufficiently large  $k$ , the complete sequence of iterates  $\{x_k\}$  ultimately converges to  $x_*$ . We now show that there is a neighbourhood of  $x_*$  in which all iterations are successful.

Suppose first that

$$\|n_k\|_2 \geq \frac{\chi\xi^N}{\xi} \|s_k\|_2. \quad (15.4.96)$$

Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this, (15.4.58), (15.4.64), and (15.4.96) give that

$$\delta m_k \geq \nu \sigma_k \delta m_k^N \geq \nu \sigma_k \kappa_{\text{bon}} \|n_k\|_2 \geq \kappa_6 \|s_k\|_2$$

for all sufficiently large  $k$ , where  $\kappa_6 \stackrel{\text{def}}{=} \nu \sigma_{\max} \kappa_{\text{bon}} \chi \xi^N / \xi$ . Combining this with (15.4.32) and (15.4.34) gives

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{\xi^2 \kappa_0 (1 + \sigma_{\max})}{\kappa_6} \Delta_k.$$

Hence the iteration will turn out to be very successful for any

$$\Delta_k \leq \frac{(1 - \eta_2) \kappa_6}{\xi^2 \kappa_0 (1 + \sigma_{\max})} \quad (15.4.97)$$

and (15.4.26) implies that the radius will not decrease in this case.

So now suppose that

$$\|n_k\|_2 < \frac{\chi\xi^N}{\xi} \|s_k\|_2 \quad (15.4.98)$$

but that  $x_k + s_k$  is unsuccessful. Then, it follows from (15.4.34) and (15.4.98) that (15.4.76) holds, and thus, by design, a second-order correction will be computed. If we return to (15.4.78), we see that the standard SQP step—if it is allowed—is

$$t_k^S = -N(x_k)(H^N(x_k))^{-1} g^N(x_k).$$

The second-order sufficiency condition AO.3 implies that  $\|(H^N(x_k))^{-1}\| \leq \kappa_{\text{uhn}}$  for some constant  $\kappa_{\text{uhn}} > 0$  for all  $x_k$  sufficiently close to  $x_*$ . Hence the above and AA.12 give the bound

$$\|t_k^S\|_2 \leq \kappa_{\text{bns}} \kappa_{\text{uhn}} \|g^N(x_k)\|_2.$$

If the standard SQP step is taken,  $t_k = t_k^S$ . Otherwise,

$$\|t_k\| \leq \xi^T \Delta_k < \|t_k^N\|$$

since, by assumption, the standard SQP step is taken whenever it lies within the trust region. Thus in both cases

$$\|t_k\|_2 \leq \kappa_{\text{bns}} \kappa_{\text{uhn}} \|g^N(x_k)\|_2 \quad \text{and} \quad \|t_k\| \leq \xi^T \Delta_k. \quad (15.4.99)$$

But then Lemma 15.4.9 and assumption AA.1i, together with (15.4.23) and (15.4.99), imply that

$$\begin{aligned}\delta m_k &\geq \kappa_{\text{btm}} \kappa_{\text{tmd}} \|g^N(x_k)\|_2 \min \left[ \frac{\xi^T \Delta_k}{\|N(x_k)\|_2}, \frac{\|g^N(x_k)\|_2}{\beta_k^T} \right] \\ &\geq \frac{\kappa_{\text{btm}} \kappa_{\text{tmd}}}{\kappa_{\text{bns}} \kappa_{\text{uhn}}} \|t_k\|_2 \min \left[ \frac{\|t_k\|_2}{\kappa_{\text{bns}}}, \frac{\|t_k\|_2}{\kappa_{\text{bns}} \kappa_{\text{uhn}} \beta_k^T} \right] \\ &= \frac{\kappa_{\text{btm}} \kappa_{\text{tmd}}}{\kappa_{\text{bns}}^2 \kappa_{\text{uhn}}} \min \left[ 1, \frac{1}{\kappa_{\text{uhn}} \beta_k^T} \right] \|t_k\|_2^2\end{aligned}\quad (15.4.100)$$

for all sufficiently large  $k$ . But if (15.4.98) holds,

$$\|s_k\|_2 \leq \|n_k\|_2 + \|t_k\|_2 \leq \chi \frac{\xi^N}{\xi} \|s_k\|_2 + \|t_k\|_2;$$

that is,

$$\|t_k\|_2 \geq \left( 1 - \chi \frac{\xi^N}{\xi} \right) \|s_k\|_2,$$

and hence (15.4.100) may be expressed as

$$\delta m_k \geq \kappa_{\text{bos}} \|s_k\|_2^2$$

for some constant  $\kappa_{\text{bos}} > 0$ . If the standard SQP step is taken, (15.4.34) and (15.4.83) show that (15.4.77) holds for all  $x_k$  sufficiently close to  $x_*$ . If the standard step is not taken, (15.4.87) implies that

$$\|s_k^{\text{CS}}\| \leq \kappa_2 \|s_k\|^2$$

for some  $\kappa_2 > 0$ , and thus again that (15.4.77) holds so long as

$$\|s_k\| \leq \Delta_k \leq \frac{\eta^{\text{CS}}}{\kappa_2}. \quad (15.4.101)$$

Thus in both cases, Step 3b of the algorithm will be executed for all  $x_k$  sufficiently close to  $x_*$  and  $s_k$  satisfying (15.4.101). Since Lemmas 15.4.13 and 15.4.14 show that (15.4.84) holds for both full and partial second-order corrections, we deduce that there is a  $\Delta_{\max}$  such that

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq 1 - \eta_2$$

for all steps  $\|s_k\| \leq \Delta_{\max}$  and  $x_k$  sufficiently close to  $x_*$ . Thus, as before, the iteration will turn out to be very successful if the radius is below some fixed value. Once again, the radius will not decrease in this case.

The final case is when (15.4.98) is satisfied and  $x_k + s_k$  is successful. In this case, (15.4.95) requires that the radius will either be unaltered or perhaps increased—this is why we needed to strengthen (15.4.26).

We have thus seen that all iterations will ultimately be successful, and the radius will be bounded away from zero. If  $s_k$  is accepted, the Q-superlinear and possible Q-quadratic convergence follow from Theorem 15.2.1. If this step is rejected, the fully

corrected step must be accepted, and the Q-superlinear and possible Q-quadratic convergence in this case follow from (15.3.45).  $\square$

In practice, we would not really expect to compute a second-order correction every time the original step gives an insufficient decrease and (15.4.76) is satisfied. In particular, when we are far from a critical point, there is perhaps little incentive to do so. However, it has been noted in practice that second-order corrections sometimes have beneficial effects even in these circumstances. We also note that, since a second SQP step is a particular full correction, the actual rate of convergence may be faster than Q-quadratic, but that this depends on the computation of additional derivatives—in addition to the constraint values  $c(x_k + s_k)$  that must necessarily be computed for any second-order correction—at the intermediate point  $x_k + s_k$ . Finally, we note that, as in Section 15.3.2.3, we need to guarantee that  $y_k$  converges to  $y_*$ , and thus we may need to ensure that  $y_k$  is close to  $y_*$  at some appropriate stage by, for instance, computing  $y_k$  as the least-squares Lagrange multiplier approximation at  $x_k$ ; thereafter, the SQP multiplier estimates will suffice.

### Notes and References for Subsection 15.4.1.5

The algorithm and results in this section are a slight generalization of those given by Byrd, Schnabel, and Shultz (1987), albeit for a different merit function. Byrd, Schnabel, and Shultz (1987) swap the restriction (15.4.94) on the second-order correction for an assumption that the limit points are first-order critical.

#### 15.4.1.6 Convergence to Second-Order Critical Points

In Section 6.6, we saw that it is relatively straightforward to extend the first-order global convergence theory of trust-region methods for unconstrained minimization to show convergence to second-order critical points. The basic idea there was to insist that the calculated step was capable of moving away from a saddle point, and this necessarily required that some estimate of an eigenvector for the leftmost eigenvalue of the Hessian be used.

Consider an idealized situation in which we start from a first-order critical point  $x$  for (15.1.2), that the reduced Hessian  $H^N(x) = N^T(x)H(x, y(x))N(x)$  is indefinite, and that we compute a unit direction  $v = N(x)v^N$  for which  $\langle v, H(x, y(x))v \rangle$  is smallest (that is, most negative). The obvious question is then: Can we ensure that our merit function will decrease for all sufficiently small steps along  $v$ ; that is, can we be sure that there is an  $\alpha_{\min} > 0$  for which  $\phi(x + \alpha v, \sigma) < \phi(x, \sigma)$  for all  $\alpha \in (0, \alpha_{\min}]$ ? Perhaps surprisingly, the answer may be no.

For consider the problem

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \quad f(x) = \frac{1}{2}x_1^2 + 2x_2 \quad \text{subject to} \quad c(x) = x_1^2 + x_2^2 - 1 = 0, \quad (15.4.102)$$

which has a nonminimizing (in fact, maximizing) first-order critical point at  $x = (0, 1)^T$  for which  $y(x) = 1$ . Given the null-space basis  $(1, 0)^T$  at  $x$ ,  $H^N(x) = -1$ , and the

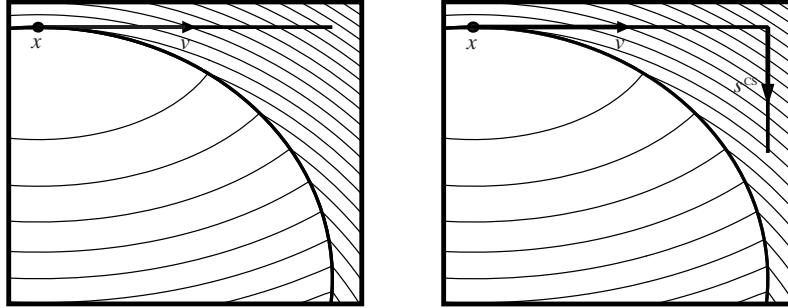


Figure 15.4.2: The Byrd–Coleman–Conn–Schnabel–Shultz effect for the problem (15.4.102) with the  $\ell_1$  penalty function. In the left-hand picture, notice how the merit function increases if we step from the constrained maximizer  $x$  to the point  $x + v$ . The beneficial effects of a second-order correction  $s^{CS}$  of the form (15.3.19) on this step are illustrated in the right-hand picture.

desired direction of negative curvature is  $v = (1, 0)^T$ . But any step in this direction increases both  $f(x)$  and  $|c(x)|$ , and thus  $\phi(x, \sigma)$  cannot decrease along  $x + \alpha v$  for any  $\alpha$ . Thus, if we only allow steps along linearized constraints, we are doomed never to escape from such a point. This “second-order Maratos” effect is, like its namesake, due to the curvature of the constraints. We shall refer to this as the *Byrd–Coleman–Conn–Schnabel–Shultz*, or *BCCSS*, effect, in honour of its discoverers. We illustrate the BCCSS effect in Figure 15.4.2.

To avoid the BCCSS effect, we note that there is (of course) a feasible descent path away from  $(0, 1)^T$  along the arc  $x_1^2 + x_2^2 - 1 = 0$ . Indeed, all other points on the constraint reduce the merit function. Thus, just as we did with the Maratos effect, we may hope to avoid the BCCSS effect by correcting for the constraint curvature after we have made a step along the linearized constraint. That is, we should use a suitable second-order correction. Once again, see Figure 15.4.2. Notice that second-order corrections may now prove to be useful for two reasons, one related to the kind of critical point we are seeking and one related to the rate of convergence to such a point, which we discussed in the previous section.

We first describe how we intend to deal with negative curvature. Since curvature plays no part in the calculation of the normal step, we shall restrict our attention to the tangential step. Just as we did in Section 6.6.3, we shall add an extra requirement to the computation of this step. Formally, we shall ask that

**AA.2d** if  $\tau_k = \lambda_{\min}[H^N(x_k)] < 0$ , the tangential step  $t_k$  gives a model reduction

$$\delta m_k^T \stackrel{\text{def}}{=} m^T(x_k, H_k, 0) - m^T(x_k, H_k, t_k) \geq -\kappa_{\text{sod}} \tau_k \Delta_k^2$$

for some constant  $\kappa_{\text{sod}} \in (0, \frac{1}{2})$ .

Such a decrease is guaranteed by Theorem 6.6.1 at the eigenpoint for the model  $m^T$  and may be viewed as simply requiring that the tangential model be minimized in some

direction that has a nontrivial component along an eigenvector corresponding to the leftmost eigenvalue of  $H^N(x_k)$  if this eigenvalue is negative.

As we saw in our example, this cannot be enough for us to prove convergence to a second-order critical point, and for this we may need to compute a second-order correction to account for constraint curvature. For simplicity, the correction we shall choose is exactly the same partial correction  $s_k^{PC}$  we considered in Section 15.4.1.5. In particular, we shall require that  $s_k^{PC}$  satisfy (15.4.85), where the component  $p_k$  satisfies (15.4.82). Of course, we also need to know when to use a second-order correction. Since the curvature is likely only to become crucial when the constraints are already satisfied up to first order, we shall only attempt a correction when the original step  $s_k$  has not provided sufficient decrease in the merit function (i.e., if  $\rho_k < \eta_1$ ) and when the normal component  $n_k$  satisfies (15.4.76), just as we did before.

**Algorithm 15.4.4: A second-order variant of Algorithm 15.4.3**

The same as Algorithm 15.4.3, except that Step 1b is replaced by

**Step 1b: Tangential step calculation.** Define a model  $m_k^T$  and compute a step  $t_k$  for which  $\|t_k\| \leq \xi^T \Delta_k$  is satisfied and AA.1i and AA.2d hold.

Of course, the extra step condition AA.2d does not change the conclusions of Theorem 15.4.12, and in particular any limit point is at least a first-order critical point. We now show that this extra step condition allows us to deduce that the limit point is actually a strong second-order critical point.

**Theorem 15.4.16** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.4 have a finite number of limit points  $(x_*^i, y_*^i)$ ,  $1 \leq i \leq \ell$ , at which AO.1b holds. Suppose further that AW.1b, AW.1c, AW.1d, AM.4j, AA.12, and AA.13 hold. Finally, suppose that  $H_k = H(x_k, y_k)$ . Then at least one of the  $x_*^i$  is a strong second-order critical point for the problem (15.1.2).

**Proof.** If  $x_*^i$  is a limit point of  $\{x_k\}$ , Theorem 15.4.12 shows that it is first-order critical. Suppose now that

$$\tau_*^i = \lambda_{\min}[N^T(x_*^i)H(x_*^i, y_*^i)N(x_*^i)] < 0 \quad (15.4.103)$$

for all  $1 \leq i \leq \ell$ . Then AW.1b implies that there is some neighbourhood  $\mathcal{N}_*^i$  of  $(x_*^i, y_*^i)$  for which

$$\lambda_{\min}[N^T(x)H(x, y)N(x)] \leq \frac{1}{2}\tau_*^i < 0$$

for all  $(x, y) \in \mathcal{N}_*^i$ . Finally, let

$$\tau_* = \min_{1 \leq i \leq \ell} \tau_*^i \text{ and } \mathcal{N}_* = \bigcup_{i=1}^{\ell} \mathcal{N}_*^i.$$

Note that this implies that there is a  $k_0 \geq 0$  such that all  $(x_k, y_k) \in \mathcal{N}_*$  for  $k \geq k_0$ , and that

$$\lambda_{\min}[N^T(x)H(x, y)N(x)] \leq \frac{1}{2}\tau_* < 0$$

for all  $(x, y) \in \mathcal{N}_*$ .

Suppose that

$$\|n_k\|_2 > \chi\xi^N\Delta_k. \quad (15.4.104)$$

Then Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this, (15.4.58), (15.4.64), and (15.4.104) give that

$$\delta m_k \geq \nu\sigma_k\delta m_k^N \geq \nu\sigma_k\kappa_{\text{bon}}\|n_k\|_2 \geq \kappa_7\Delta_k$$

for all sufficiently large  $k$ , where  $\kappa_7 \stackrel{\text{def}}{=} \nu\sigma_{\max}\kappa_{\text{bon}}\chi\xi^N$ . Combining this with (15.4.33) gives

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{\kappa_{\text{ufm}}(1 + \sigma_{\max})}{\kappa_7} \Delta_k,$$

and hence the iteration will turn out to be very successful, (15.4.95) implies that the radius will not decrease, and

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k) \geq \eta_2\delta m_k \geq \eta_2\kappa_7\Delta_k \geq \eta_2\kappa_7\Delta_k^2 \quad (15.4.105)$$

for all

$$\Delta_k \leq \min \left[ 1, \frac{(1 - \eta_2)\kappa_7}{\kappa_{\text{ufm}}(1 + \sigma_{\max})} \right].$$

Suppose instead that

$$\|n_k\|_2 \leq \chi\xi^N\Delta_k. \quad (15.4.106)$$

Since  $(x_k, y_k) \in \mathcal{N}_*$ ,  $\tau_k \leq \frac{1}{2}\tau_* < 0$ , and Lemma 15.4.9 and assumption AA.2d together imply that

$$\delta m_k \geq \kappa_8\Delta_k^2, \text{ where } \kappa_8 = -\frac{1}{2}\kappa_{\text{btm}}\kappa_{\text{sod}}\tau_* > 0. \quad (15.4.107)$$

Thus, if the step  $s_k$  is successful,

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k) \geq \eta_1\kappa_8\Delta_k^2, \quad (15.4.108)$$

and again (15.4.95) implies that the radius does not decrease. If the step is unsuccessful and since (15.4.106) is precisely the condition (15.4.76), a partial second-order correction  $s_k^{\text{CS}} = s_k^{\text{PC}}$  will be contemplated. Since in this case (15.4.87) implies that

$$\|s_k^{\text{CS}}\| \leq \kappa_2\|s_k\|^2$$

for some  $\kappa_2 > 0$ , it follows that (15.4.77) holds so long as

$$\|s_k\| \leq \Delta_k \leq \frac{\eta^{\text{CS}}}{\kappa_2}. \quad (15.4.109)$$

Thus Step 3b of the algorithm will be executed for all  $s_k$  satisfying (15.4.109). For the particular correction  $s_k^{\text{CS}} = s_k^{\text{PC}}$ , Lemma 15.4.14, (15.4.34), and (15.4.107) imply that

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq 1 - \eta_2 \quad (15.4.110)$$

for all steps  $\|s_k\| \leq \Delta_{\max}^{\text{C}}$ , some  $\Delta_{\max}^{\text{C}} > 0$ , and all  $x_k$  sufficiently close to  $x_*$ . Thus, the second-order correction will again be very successful, (15.4.95) implies that the radius does not decrease, and (15.4.107) and (15.4.110) show that

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) \geq \eta_2 \delta m_k \geq \eta_2 \kappa_8 \Delta_k^2. \quad (15.4.111)$$

Since all iterates are ultimately within  $\mathcal{N}_*$ , the trust-region radii satisfy

$$\Delta_k \geq \Delta_{\max} \quad (15.4.112)$$

for some  $\Delta_{\max} > 0$  and all  $k \geq 0$ . Hence there must be an infinite number of successful iterations, and on each of these (15.4.105), (15.4.108), (15.4.111), and (15.4.112) show that

$$\phi(x_k, \sigma_{\max}) - \phi(x_{k+1}, \sigma_{\max}) \geq \kappa_9 \Delta_{\max}^2$$

for some  $\kappa_9 > 0$ . But this is impossible since AW.1d implies that  $\phi(x, \sigma_{\max})$  is bounded from below. Hence our assumption (15.4.103) must be false,

$$\lambda_{\min}[N^T(x_*^i)H(x_*^i, y_*^i)N(x_*^i)] \geq 0$$

for some  $1 \leq i \leq \ell$ , and thus  $x_*^i$  is a strong second-order critical point for (15.1.2).

□

## Notes and References for Subsection 15.4.1.6

The BCCSS effect was first observed by Coleman and Conn (1980) at infeasible first-order critical points. The example given here, at a feasible first-order critical point, was given by Byrd, Schnabel, and Shultz (1987). Once again, the results here are modelled on those given by Byrd, Schnabel, and Shultz (1987).

### 15.4.2 Byrd–Omojokun-like Approaches

As we have said, the aim of the normal-step component in a composite-step method is to move towards feasibility of the linearized constraints (15.2.9b) and (15.2.9c). Rather

than achieve this as we did in Section 15.4.1, another possibility is to compute  $n_k$  to approximately

$$\underset{n \in \mathbb{R}^n}{\text{minimize}} \quad \|A(x_k)n + c(x_k)\| \quad \text{subject to} \quad \|n\| \leq \xi^N \Delta_k \quad (15.4.113)$$

for some  $0 < \xi^N < 1$ . Of course, this problem may have a large number of solutions, but we shall not be concerned at this stage with which is chosen—the minimum-norm solution will give a component that is normal to  $t_k$ . The choice of norm also plays an important role. Early methods used the  $\ell_2$  norm for both objective and trust region, although using a polyhedral norm, such as the  $\ell_1$  or  $\ell_\infty$  norm, has the possible advantage that the subproblem is then a linear or quadratic programming problem.

Having found  $n_k$ , the tangential component is chosen so that the linearized constraint infeasibility remains constant and the flexibility in  $t_k$  is used to reduce a suitable model within the true trust region. (If there is a feasible point for  $A(x)s + c(x) = 0$  within the shrunken trust region  $\|s\| \leq \xi^N \Delta$  and the linearized constraint infeasibility remains constant, the composite step will also be feasible for this linearization.) This is achieved exactly as in Section 15.4.1 by finding  $t_k$  to approximately

$$\underset{t \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle t, H_k t \rangle + \langle g(x_k) + H_k n_k, t \rangle \quad (15.4.114a)$$

$$\text{subject to} \quad A(x_k)t = 0 \quad (15.4.114b)$$

$$\text{and} \quad \|n_k + t\| \leq \Delta_k \quad (15.4.114c)$$

for some appropriate approximation,  $H_k$ , to the Hessian of the Lagrangian. This problem always has a feasible point  $t = 0$ .

Having set the scene, we shall again build and analyse an algorithm based on these ideas. For the purposes of exposition, we shall consider a particular case, namely, where the objective function in (15.4.113) and the trust region for both subproblems are specified in the  $\ell_2$  norm. This analysis can easily be extended to other (perhaps weighted) trust-region norms. Fortunately, much of what we require has already been developed in detail in Section 15.4.1. We shall consider the same  $\ell_2$  penalty function as our merit function, and the same model (15.4.6) as a suitable approximation, as we used in Section 15.4.1.1. The penalty parameter will always be chosen to ensure that (15.4.7) holds. We now discuss exactly how we shall choose our steps.

### 15.4.2.1 The Normal and Tangential Steps

As we have emphasized throughout this book, we should not expect to solve the normal and tangential step subproblems very accurately, particularly when the problem involves a large number of unknowns. Once again, we must consider what it means to find an approximate solution to our subproblems. By now, we hope that the reader will have already anticipated that the requirements we shall impose are based upon the calculation of appropriate Cauchy points.

Consider first the normal step. The gradient of the model

$$m^N(x, n) = \|c(x) + A(x)n\|_2 \quad (15.4.115)$$

is

$$\nabla_n m^N(x, n) = \frac{A^T(x)(c(x) + A(x)n)}{\|c(x) + A(x)n\|_2}$$

so long as  $c(x) + A(x)n \neq 0$ . In particular, if  $c(x) \neq 0$ , the steepest-descent step for the model at  $n = 0$  is in the direction  $-A^T(x)c(x)$ . The Cauchy step  $n^C$  for the normal step subproblem (15.4.113) is then defined to be the minimizer of the model (15.4.115) in the steepest-descent direction within the trust region  $\|n\| \leq \xi^N \Delta$ ; that is,

$$n^C = -\alpha^C A^T(x)c(x), \quad \text{where} \quad (15.4.116)$$

$$\alpha^C = \arg \min_{\alpha \geq 0} \|c(x) - \alpha A(x)A^T(x)c(x)\|_2 \quad \text{subject to } \alpha \|A^T(x)c(x)\|_2 \leq \xi^N \Delta.$$

For such a step, we have the following result.

**Lemma 15.4.17** Suppose that  $n^C$  is the Cauchy step for (15.4.113). Then

$$m^N(x, 0) - m^N(x, n^C) \geq \frac{1}{2} \frac{\|A^T(x)c(x)\|_2}{\|c(x)\|_2} \min \left[ \xi^N \Delta, \frac{\|A^T(x)c(x)\|_2}{\beta^N} \right],$$

where  $\beta^N = 1 + \|A^T(x)A(x)\|_2$ .

**Proof.** In view of Corollary 6.3.2 (p. 127) applied to one-half of the square of the model (15.4.115), we have that

$$\frac{1}{2} (\|c(x)\|_2^2 - \|c(x) + A(x)n^C\|_2^2) \geq \frac{1}{2} \|A^T(x)c(x)\|_2 \min \left[ \xi^N \Delta, \frac{\|A^T(x)c(x)\|_2}{\beta^N} \right]. \quad (15.4.117)$$

But

$$\begin{aligned} & \|c(x)\|_2^2 - \|c(x) + A(x)n^C\|_2^2 \\ &= (\|c(x)\|_2 - \|c(x) + A(x)n^C\|_2) (\|c(x)\|_2 + \|c(x) + A(x)n^C\|_2) \\ &\leq 2\|c(x)\|_2 (\|c(x)\|_2 - \|c(x) + A(x)n^C\|_2) \end{aligned} \quad (15.4.118)$$

using  $\|c(x) + A(x)n^C\|_2 \leq \|c(x)\|_2$  and the definition of  $n^C$ . Combining (15.4.117) and (15.4.118) gives the required result.  $\square$

In view of Lemma 15.4.17, and since we may expect to decrease the model at least a fraction of what we would achieve at the Cauchy point, we shall require a reduction on the normal-step model of the following form.

**AA.1j** For all  $k$ , the normal step  $n_k$  gives a model reduction

$$\begin{aligned} \delta m_k^N &\stackrel{\text{def}}{=} m^N(x_k, 0) - m^N(x_k, n_k) \\ &\geq \kappa_{\text{dns}} \frac{\|A^T(x_k)c(x_k)\|_2}{\|c(x_k)\|_2} \min \left[ \xi^N \Delta_k, \frac{\|A^T(x_k)c(x_k)\|_2}{\beta_k^N} \right] \end{aligned}$$

for some constant<sup>255</sup>  $\kappa_{\text{dns}} \in (0, 1)$ , where  $\beta_k^{\text{N}} = 1 + \|A^T(x_k)A(x_k)\|_2$ .

Although as we have said, it is not necessary for the normal step to be actually normal to the tangential step, it will turn out that we cannot allow it to be too un-normal. Suppose that  $R_k$  and  $N_k$  are matrices whose columns form bases for the range- and null-spaces of  $A(x_k)$  and that the smallest singular values of these matrices are uniformly bounded away from zero, while their largest singular values are uniformly bounded from above. Then we can decompose

$$n_k = R_k n_k^{\text{R}} + N_k n_k^{\text{N}}, \quad (15.4.119)$$

and we shall require that

**AA.1k** there is a constant<sup>256</sup>  $\kappa_{\text{drs}} > 0$  such that

$$\|n_k^{\text{N}}\|_2 \leq \kappa_{\text{drs}} \|n_k^{\text{R}}\|_2 \quad (15.4.120)$$

for all  $k$ , where  $n_k$  is decomposed according to (15.4.119), and  $R_k$  and  $N_k$  are as described above. In addition,  $n_k^{\text{R}} = 0$  whenever  $A^T(x_k)c(x_k) = 0$ .

This condition ensures that any approximate solution to (15.4.113) we choose has a nontrivial component in the range-space of  $A$ . In particular,

$$n_k = 0 \quad \text{if } A^T(x_k)c(x_k) = 0. \quad (15.4.121)$$

Such a condition is important, since there may be other (nonzero) solutions to (15.4.113) if  $A(x_k)$  is rank deficient.

We should mention that, although we earlier suggested that  $0 < \xi^{\text{N}} < 1$ , in theory it does not actually matter how large an upper bound is used. However, for consistency with the analysis of Section 15.4.1, we henceforth only require that  $0 < \xi^{\text{N}} \leq 1$ .

From a practical point of view, finding a normal step that satisfies AA.1j and AA.1k is straightforward. Obvious examples are the Cauchy step and the (minimum-norm) model minimizers—it is simple to show that condition (15.4.120) is satisfied, since the normal steps in these cases lie entirely in the range of  $A^T(x_k)$ . More importantly, methods like the Steihaug–Toint truncated conjugated gradient method (Section 7.5.1) and the generalized Lanczos trust-region method (Section 7.5.4) applied to the subproblem (15.4.113) (squaring the model objective function) both produce a suitable step; once again, it is straightforward to show that the steps generated by such methods stay in the range of  $A^T(x_k)$  (see in particular Section 5.3).

Given the normal step, the tangential step is computed exactly as we did in Section 15.4.1.2. We shall require that this step satisfy AA.1i and assume that any computed null-space basis satisfies AA.12. Methods for computing a suitable tangential step have already been considered in Section 15.4.1.2.

<sup>255</sup>“dns” stands for “decrease of the model along the normal step”.

<sup>256</sup>“drs” stands for “dominant range space”.

### 15.4.2.2 The Byrd–Omojokun Algorithm

We are now in a position to state our complete algorithm, named after its co-founders. Fortunately for us, the algorithm is essentially the same as Algorithm 15.4.1.

**Algorithm 15.4.5: The Byrd–Omojokun composite-step algorithm**

The same as Algorithm 15.4.1 except that Step 1a is replaced by

**Step 1a: Normal step calculation.** Define a model  $m_k^N$  and compute a step  $n_k$  for which  $\|n_k\| \leq \xi^N \Delta_k$  is satisfied and AA.1j and AA.1k hold.

As before, reasonable values for the parameters in Algorithm 15.4.5 might be

$$\eta_1 = 0.01, \eta_2 = 0.75, \gamma_1 = \gamma_2 = \frac{1}{2}, \xi^N = \xi^T = 0.8, \nu = 0.0001, \tau_1 = 2, \text{ and } \tau_2 = 1,$$

but other values may be preferred.

### 15.4.2.3 Convergence Analysis

We now consider the global convergence behaviour of Algorithm 15.4.5. As always, our aim is to show that, under ideal conditions, the iterates converge to a first-order critical point for problem (15.1.2), that is, that there is a point for which (15.4.28) holds. We shall see, however, that this is not always possible, most especially when there is no feasible point. Under these circumstances, nonetheless, we shall show that the algorithm converges to a critical point for a related “least-infeasibility” problem. Our analysis follows very closely the one presented in Section 15.4.1.4 for Algorithm 15.4.1, and indeed, we can merely recycle a number of the results obtained there.

Throughout our analysis, we shall continue to require that AW.1c, and hence the bounds (15.4.30) and (15.4.31), hold. These then imply that the bound between the model and merit function at  $x_k + s_k$  given by Lemma 15.4.3 is true. We now show that the analog of Lemma 15.4.4, which shows that progress may always be made from a noncritical point, continues to hold in almost all cases.

**Lemma 15.4.18** Assume that AW.1c, AM.4j, AA.12, and AA.13 hold. Suppose that at least one of  $A^T(x_k)c(x_k)$  or  $N^T(x_k)g(x_k)$  is nonzero at the point  $x_k$  generated by Algorithm 15.4.5. Then there exists a  $\Delta_k^0 > 0$  such that  $\rho_k \geq \eta_1$  if  $\Delta_k \in (0, \Delta_k^0)$ .

**Proof.** The proof is similar to that of Lemma 15.4.4. There are two cases to consider, namely,  $A^T(x_k)c(x_k) = 0$  and  $A^T(x_k)c(x_k) \neq 0$ .

Firstly, suppose that  $A^T(x_k)c(x_k) = 0$ . In this case, (15.4.121), which follows from AA.1k, requires that  $n_k = 0$ , and thus that  $s_k = t_k$ , where  $t_k$  satisfies AA.1i. Then, reasoning exactly as in the proof of Lemma 15.4.4, we deduce that  $\rho_k \geq \eta_1$ .

So now suppose instead that  $A^T(x_k)c(x_k)$ , and thus  $c(x_k)$ , is nonzero. Then Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this and AA.1j show that

$$\begin{aligned}\delta m_k &\geq \kappa_{\text{dns}} \nu \sigma_k \frac{\|A^T(x_k)c(x_k)\|_2}{\|c(x_k)\|_2} \min \left[ \xi^N \Delta_k, \frac{\|A^T(x_k)c(x_k)\|_2}{\beta_k^N} \right] \\ &\geq \kappa_{\text{dns}} \nu \sigma_k \gamma_k \min \left[ \xi^N \Delta_k, \frac{\gamma_k \|c(x_k)\|_2}{\beta_k^N} \right],\end{aligned}\quad (15.4.122)$$

while the curvature  $\beta_k^N$  in AA.1j satisfies the bound

$$\beta_k^N \leq 1 + \kappa_k^2, \quad (15.4.123)$$

where

$$\gamma_k \stackrel{\text{def}}{=} \|A^T(x_k)c(x_k)\|_2 / \|c(x_k)\|_2 \neq 0$$

and

$$\kappa_k \stackrel{\text{def}}{=} \|A(x_k)\|.$$

Let

$$\Delta_k^0 = \gamma_k \min \left[ \frac{\|c(x_k)\|_2}{\xi^N(1 + \kappa_k^2)}, \frac{(1 - \eta_1)\kappa_{\text{dns}}\xi^N\nu\sigma_{-1}}{\kappa_{\text{ufm}}(1 + \sigma_{-1})} \right]. \quad (15.4.124)$$

If  $\Delta_k \leq \Delta_k^0$ , (15.4.122), (15.4.123), and (15.4.124) show that

$$\delta m_k \geq \kappa_{\text{dns}} \nu \xi^N \sigma_k \gamma_k \Delta_k,$$

which combines with (15.4.33) to give

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{\kappa_{\text{ufm}}(1 + \sigma_k)}{\kappa_{\text{dns}} \nu \xi^N \sigma_k \gamma_k} \Delta_k. \quad (15.4.125)$$

But then the fact that  $\Delta_k \leq \Delta_k^0$  and (15.4.124) and the fact that  $\{\sigma_k\}$  is nondecreasing combine with (15.4.125) to show that once again  $\rho_k \geq \eta_1$ , which proves the lemma.  $\square$

Notice that we have not shown that progress may always be made from a noncritical point, since Lemma 15.4.18 says nothing about the case where both  $A^T(x_k)c(x_k)$  and  $N^T(x_k)g(x_k)$  are zero. Indeed, the algorithm cannot make progress from such a point without further modification (see Section 15.4.2.5). In order to ensure that a limit point is first-order critical, we shall require stronger assumptions. Before we do this, we first consider the sequence  $\{A^T(x_k)c(x_k)\}$  generated by the algorithm.

We require the following variant of assumption AW.1d.

**AW.1f** The function  $f(x)$  is uniformly bounded from below, while the functions  $g(x)$ ,  $c(x)$ , and  $A(x)$  are uniformly bounded at all points computed by the algorithm.

This assumption is implied, for instance, by AI.1, but is marginally weaker. As before, the intention of such an assumption is purely to ensure that the merit function and its derivatives are well behaved at points encountered by the algorithm. In particular, AW.1f again implies (15.4.41), which in turn implies that Lemma 15.4.5 holds.

The first main result we obtain is to show that any limit point must, at the very least, be a critical point for the square of the infeasibility  $\|c(x)\|_2^2$ . This does not, of course, imply that such a limit point is feasible, but is an indication that it is at least locally as close to the feasible point as is possible. Of course, this result is weaker than Theorem 15.4.6, but reflects the fact that the requirements AA.1g and AA.1h on the iterates for Algorithm 15.4.1 are stronger than the equivalent AA.1i required for the current algorithm.

**Theorem 15.4.19** Suppose that AW.1c and AW.1f hold. Then

$$\lim_{k \rightarrow \infty} A^T(x_k)c(x_k) = 0.$$

**Proof.** Let  $\theta(x) \stackrel{\text{def}}{=} \|A^T(x)c(x)\|_2$  be the norm of the gradient of  $\frac{1}{2}\|c(x)\|_2^2$ . It follows from AW.1c and AW.1f that it is Lipschitz continuous. Since the result is trivial if  $\theta(x_\ell) = 0$  for all sufficiently large  $\ell$ , consider an arbitrary infeasible iterate  $x_\ell$ , that is, one for which  $\theta(x_\ell) > 0$ . Our first aim is to show that Algorithm 15.4.5 cannot stall at such a point.

The Lipschitz continuity of  $\theta$  implies that

$$|\theta(x) - \theta(x_\ell)| \leq \gamma\|x - x_\ell\|_2 \quad (15.4.126)$$

for some  $\gamma > 0$  and all  $x$ . Thus (15.4.126) holds for all  $x$  in the open ball

$$\mathcal{O}_\ell \stackrel{\text{def}}{=} \left\{ x \mid \|x - x_\ell\|_2 < \frac{\theta(x_\ell)}{2\gamma} \right\}. \quad (15.4.127)$$

It then follows immediately from (15.4.126) and (15.4.127) that  $|\theta(x) - \theta(x_\ell)| < \frac{1}{2}\theta(x_\ell)$  and thus that

$$\theta(x) > \frac{1}{2}\theta(x_\ell), \quad (15.4.128)$$

which in turn ensures that  $c(x)$  is bounded away from zero for any  $x \in \mathcal{O}_\ell$ . Now, arguing as we did in the second half of Lemma 15.4.18, Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this, AA.1j, (15.4.128), and the bound  $\|c(x_k)\|_2 \leq c_{\max}$  that is implied by AW.1f show that

$$\delta m_k \geq \frac{\kappa_{\text{dns}}\nu\sigma_k\theta(x_k)}{\|c(x_k)\|_2} \min \left[ \xi^N \Delta_k, \frac{\theta(x_k)}{\beta_k^N} \right] \geq \frac{\kappa_{\text{dns}}\nu\sigma_k\theta(x_\ell)}{2c_{\max}} \min \left[ \xi^N \Delta_k, \frac{\theta(x_\ell)}{2\beta_k^N} \right] \quad (15.4.129)$$

for any  $x_k \in \mathcal{O}_\ell$ . Let

$$\Delta_k^0 = \frac{1}{2}\theta(x_\ell) \min \left[ \frac{1}{\beta_{\max}\xi^N}, \frac{(1-\eta_2)\kappa_{\text{dns}}\xi^N\nu\sigma_{-1}}{\kappa_{\text{ufm}}c_{\max}(1+\sigma_{-1})} \right], \quad (15.4.130)$$

where AW.1f implies that

$$\beta_k^N \leq \beta_{\max} \stackrel{\text{def}}{=} 1 + \max_i \|A(x_i)\|_2^2. \quad (15.4.131)$$

If  $\Delta_k \leq \Delta_\ell^0$ , (15.4.129), (15.4.130), and (15.4.131) show that

$$\delta m_k \geq \frac{\kappa_{\text{dns}} \xi^N \nu \sigma_k \theta(x_\ell)}{2c_{\max}} \Delta_k,$$

which combines with (15.4.33) to give

$$|\rho_k - 1| = \left| \frac{\phi(x_k + s_k, \sigma_k) - m(x_k, H_k, \sigma_k, s_k)}{\delta m_k} \right| \leq \frac{2\kappa_{\text{ufm}} c_{\max} (1 + \sigma_k)}{\kappa_{\text{dns}} \xi^N \nu \sigma_k \theta(x_\ell)} \Delta_k. \quad (15.4.132)$$

But then the facts that  $\{\sigma_k\}$  is nondecreasing and that  $\Delta_k \leq \Delta_\ell^0$  and (15.4.130) combine with (15.4.132) to show that  $\rho_k \geq \eta_2$ . Hence, so long as  $\Delta_k \leq \Delta_\ell^0$ , a very successful iteration will occur from any  $x_k \in \mathcal{O}_\ell$ .

Next let  $\mathcal{S}$  be the indices of successful iterates and suppose that all iterates  $x_{k_i}$ ,  $k_i \in \mathcal{S} \geq k_0 \stackrel{\text{def}}{=} \ell$ , remain in  $\mathcal{O}_\ell$ . Then

$$\Delta_{k_i} \geq \Delta^0 \stackrel{\text{def}}{=} \gamma_1 \min[\Delta_\ell, \Delta_\ell^0], \quad (15.4.133)$$

since, as we saw in the previous paragraph, the radius cannot fall below this value without the iteration being very successful—the subsequent radius will be no smaller. Since iteration  $k_i \in \mathcal{S}$  is successful, the bounds (15.4.43), (15.4.129), (15.4.131), and (15.4.133) give that

$$\begin{aligned} \psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) &\leq \psi(x_{k_i}, \sigma_{k_i}) - \eta_1 \frac{\delta m_{k_i}}{\sigma_{k_i}} \\ &\leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{\eta_1 \kappa_{\text{dns}} \nu \theta(x_\ell)}{2c_{\max}} \min \left[ \xi^N \Delta^0, \frac{\theta(x_\ell)}{2\beta_{\max}} \right]. \end{aligned}$$

Summing over the first  $j$  of these successful iterations, we then have that

$$\psi(x_{k_{i+j}}, \sigma_{k_{i+j}}) \leq \psi(x_{k_i}, \sigma_{k_i}) - j \frac{\eta_1 \kappa_{\text{dns}} \nu \theta(x_\ell)}{2c_{\max}} \min \left[ \xi^N \Delta^0, \frac{\theta(x_\ell)}{2\beta_{\max}} \right].$$

Since the right-hand side of this inequality can be made arbitrarily negative by increasing  $j$ ,  $\psi(x_{k_{i+j}}, \sigma_{k_{i+j}})$  must eventually be negative. However, as this contradicts (15.4.42), our hypothesis that  $x_{k_i}$ ,  $k_i \in \mathcal{S} \geq \ell$ , remain in  $\mathcal{O}_\ell$  must be false, and thus there must be a first iterate  $x_{k_j}$ ,  $j > 0$ , not in  $\mathcal{O}_\ell$ .

Consider the history of iterates between this  $x_{k_j}$  and  $x_{k_0} = x_\ell$ . Combining (15.4.43), (15.4.129), and (15.4.131)

$$\psi(x_{k_k}, \sigma_{k_k}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{\eta_1 \kappa_{\text{dns}} \nu \theta(x_\ell)}{2c_{\max}} \min \left[ \xi^N \Delta_{k_i}, \frac{\theta(x_\ell)}{2\beta_{\max}} \right] \quad (15.4.134)$$

for any  $i < k \leq j$ , since  $x_{k_i} \in \mathcal{O}_\ell$ . If

$$\Delta_{k_i} \geq \frac{\theta(x_\ell)}{2\beta_{\max} \xi^N},$$

(15.4.43) and (15.4.134) with  $k = j$  give that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{\eta_1 \kappa_{\text{dns}} \nu \theta^2(x_\ell)}{4c_{\max} \beta_{\max}} \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \kappa_{\text{dns}} \nu \theta^2(x_\ell)}{4c_{\max} \beta_{\max}}. \quad (15.4.135)$$

On the other hand, if

$$\Delta_{k_i} < \frac{\theta(x_\ell)}{2\beta_{\max} \xi^N}$$

for all  $0 \leq i \leq j$ ,

$$\psi(x_{k_{i+1}}, \sigma_{k_{i+1}}) \leq \psi(x_{k_i}, \sigma_{k_i}) - \frac{\eta_1 \kappa_{\text{dns}} \xi^N \nu \theta(x_\ell)}{2c_{\max}} \Delta_{k_i} \quad (15.4.136)$$

for each  $0 \leq i < j$ , and hence, on summing (15.4.136),

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \kappa_{\text{dns}} \xi^N \nu \theta(x_\ell)}{2c_{\max}} \sum_{i=0}^{j-1} \Delta_{k_i}. \quad (15.4.137)$$

But as  $x_{k_j} \notin \mathcal{O}_\ell$ , (15.4.34) and (15.4.127) give

$$\frac{\theta(x_\ell)}{2\gamma} \leq \|x_{k_j} - x_\ell\|_2 \leq \sum_{i=0}^{j-1} \|x_{k_{i+1}} - x_{k_i}\|_2 \leq \xi \sum_{i=0}^{j-1} \Delta_{k_i}, \quad (15.4.138)$$

and therefore (15.4.137) implies that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \frac{\eta_1 \kappa_{\text{dns}} \xi^N \nu \theta^2(x_\ell)}{4\gamma\xi c_{\max}}. \quad (15.4.139)$$

Thus in all cases (15.4.135) and (15.4.139) imply that

$$\psi(x_{k_j}, \sigma_{k_j}) \leq \psi(x_\ell, \sigma_\ell) - \kappa_1 \theta^2(x_\ell), \quad \text{where } \kappa_1 \stackrel{\text{def}}{=} \frac{\eta_1 \kappa_{\text{dns}} \nu}{4c_{\max}} \max\left(\frac{1}{\beta_{\max}}, \frac{\xi^N}{\gamma}\right). \quad (15.4.140)$$

Since  $\ell$  was arbitrary, and since Lemma 15.4.5 shows that  $\{\psi(x_k, \sigma_k)\}$  is decreasing and bounded from below, (15.4.140) shows that  $\theta(x_\ell)$  has to converge to zero.  $\square$

As we have assumed that both  $c(x_k)$  and  $A(x_k)$  are bounded, they both have limit points. If  $c_*$  and  $A_*$  are the limiting values, Theorem 15.4.19 shows that either  $c_* = 0$  or  $A_*^T$  is rank deficient (or both). In particular, if  $c_* \neq 0$ ,  $A_*^T$  must be rank deficient. In order to rule out this second possibility, we need to assume that

**AA.14** there is a constant<sup>257</sup>  $\kappa_{\text{brs}} > 1$  such that for all  $k$

$$0 < \frac{1}{\kappa_{\text{brs}}} \leq \sigma_{\min}[A(x_k)] \leq \sigma_{\max}[A(x_k)] \leq \kappa_{\text{brs}}.$$

This immediately leads to our first main convergence result.

---

<sup>257</sup>“brs” stands for “bounded conditioning of the range-space”.

**Corollary 15.4.20** Suppose that AW.1c, AW.1f, and AA.14 hold. Then

$$\lim_{k \rightarrow \infty} c(x_k) = 0.$$

**Proof.** The result follows immediately from Theorem 15.4.19 and AA.14.  $\square$

We now turn to the other required first-order criticality condition,  $N^T(x_*)g(x_*) = 0$ . Our first result is the direct analog of Lemma 15.4.7.

**Lemma 15.4.21** Suppose that AW.1c, AW.1f, AM.4j, AA.12, and AA.14 hold. Then there is a constant<sup>258</sup>  $\kappa_{\text{bon}} > 0$  such that

$$\kappa_{\text{bon}} \|n_k\|_2 \leq \delta m_k^N \leq \|c(x_k)\|_2 \quad (15.4.141)$$

for all  $k$ .

**Proof.** The result is trivially true if  $c(x_k) = 0$ , since the consequence (15.4.121) of AA.1k requires that  $n_k = 0$ . So now assume that  $c(x_k) \neq 0$ . Assumption AA.14 implies that<sup>259</sup>

$$\beta_k^N \leq \kappa_{\text{bnc}} \stackrel{\text{def}}{=} 1 + \kappa_{\text{brs}}^2,$$

while this bound, AA.1j, AA.14, and the Rayleigh-quotient inequality show that

$$\begin{aligned} \delta m_k^N &\geq \kappa_{\text{dns}} \frac{\|A^T(x_k)c(x_k)\|_2}{\|c(x_k)\|_2} \min \left[ \xi^N \Delta_k, \frac{\|A^T(x_k)c(x_k)\|_2}{\beta_k^N} \right] \\ &\geq \frac{\kappa_{\text{dns}}}{\kappa_{\text{brs}}} \min \left[ \xi^N \Delta_k, \frac{\kappa_{\text{brs}} \|c(x_k)\|_2}{\kappa_{\text{bnc}}} \right]. \end{aligned} \quad (15.4.142)$$

Now suppose that

$$\Delta_k \leq \frac{2(\kappa_{\text{brs}} + \kappa_{\text{bns}} \kappa_{\text{drs}}) \kappa_{\text{brs}}^2}{\xi^N} \|c(x_k)\|_2.$$

Then (15.4.142) implies that

$$\delta m_k^N \geq \kappa_{\text{bon}} \xi^N \Delta_k \geq \kappa_{\text{bon}} \|n_k\|_2, \quad (15.4.143)$$

where

$$\kappa_{\text{bon}} \stackrel{\text{def}}{=} \frac{\kappa_{\text{dns}}}{\kappa_{\text{brs}}} \min \left[ 1, \frac{1}{2\kappa_{\text{bnc}} \kappa_{\text{brs}} (\kappa_{\text{brs}} + \kappa_{\text{bns}} \kappa_{\text{drs}})} \right],$$

since  $\|n_k\|_2 \leq \xi^N \Delta_k$ , which is of the required form (15.4.141). On the other hand, if

$$\Delta_k > \frac{2(\kappa_{\text{brs}} + \kappa_{\text{bns}} \kappa_{\text{drs}}) \kappa_{\text{brs}}^2}{\xi^N} \|c(x_k)\|_2, \quad (15.4.144)$$

<sup>258</sup>“bon” stands for “bound on the normal step”.

<sup>259</sup>“bnc” stands for “bound on the normal-step curvature”.

(15.4.142) gives that

$$\delta m_k^N \geq \kappa_{dns} \min \left[ 2(\kappa_{brs} + \kappa_{bns} \kappa_{drs}) \kappa_{brs}, \frac{1}{\kappa_{bnc}} \right] \|c(x_k)\|_2. \quad (15.4.145)$$

However, since  $n_k$  is chosen to reduce  $\|c(x_k) + A(x_k)n\|_2$ , we directly obtain that  $\|c(x_k) + A(x_k)n_k\|_2^2 \leq \|c(x_k)\|_2^2$  and hence that

$$\|A(x_k)n_k\|_2^2 \leq -2\langle c(x_k), A(x_k)n_k \rangle \leq 2\|c(x_k)\|_2\|A(x_k)n_k\|_2,$$

which implies that

$$\|A(x_k)n_k\|_2 \leq 2\|c(x_k)\|_2. \quad (15.4.146)$$

Writing

$$n_k = A^T(x_k)n_k^R + N(x_k)n_k^N,$$

(15.4.146) and AA.14 give that

$$\frac{\|n_k^R\|_2}{\kappa_{brs}^2} \leq \|A(x_k)A^T(x_k)n_k^R\|_2 \leq 2\|c(x_k)\|_2.$$

Combining these with AA.12, AA.14, and AA.1k then leads to

$$\begin{aligned} \|n_k\|_2 &\leq \|A^T(x_k)\|_2\|n_k^R\|_2 + \|N(x_k)\|_2\|n_k^N\|_2 \\ &\leq (\kappa_{brs} + \kappa_{bns} \kappa_{drs})\|n_k^R\|_2 \\ &\leq 2(\kappa_{brs} + \kappa_{bns} \kappa_{drs})\kappa_{brs}^2\|c(x_k)\|_2. \end{aligned} \quad (15.4.147)$$

But (15.4.144) and (15.4.147) together imply that  $\|n_k\|_2 < \xi^N \Delta_k$  and thus that this  $n_k$  is indeed within the trust region. But then (15.4.145) and (15.4.147) show, once again, that

$$\delta m_k^N \geq \frac{\kappa_{dns}}{\kappa_{brs}} \min \left[ 1, \frac{1}{2\kappa_{bnc}(\kappa_{brs} + \kappa_{bns} \kappa_{drs})\kappa_{brs}} \right] \|n_k\| = \kappa_{bon}\|n_k\|_2. \quad (15.4.148)$$

Thus, as either (15.4.143) or (15.4.148) hold, and since

$$\delta m_k^N = \|c(x_k)\|_2 - \|c(x) + A(x)n\|_2 \leq \|c(x_k)\|_2,$$

the lemma is true.  $\square$

Having shown that the normal model is eventually insignificant, the remaining analysis is exactly as for Algorithm 15.4.1. In particular, so long as we replace AW.1d by AW.1f, the statements and proofs of Theorem 15.4.8 and Lemma 15.4.9 hold equally for Algorithm 15.4.5, as do Theorems 15.4.10 and 15.4.11 so long as we add assumption AA.14. For future reference, we record this as Theorem 15.4.22.

**Theorem 15.4.22** Suppose that AW.1c, AW.1f, AM.4j, AA.12, AA.13, and AA.14 hold. Then there are constants  $\sigma_{\max} > 0$  and  $\kappa_{\text{btm}} > 0$  and an index  $k_1$  such that

$$\sigma_k = \sigma_{\max} \text{ and} \quad (15.4.149)$$

$$\delta m_k \geq \kappa_{\text{btm}} \delta m_k^T \quad (15.4.150)$$

for all  $k \geq k_1$ . Moreover,

$$\lim_{k \rightarrow \infty} \|N^T(x_k)g(x_k)\| = 0.$$

To summarize, we have seen that in all cases any limit point must occur at a critical point for the sum of squares of the infeasibility (Theorem 15.4.19). This may be all we can say if the limiting constraint Jacobian is rank deficient. If the limiting Jacobian is of full rank, the first-order criticality conditions will be satisfied (Corollary 15.4.20 and Theorem 15.4.22), and moreover, the penalty parameter cannot become arbitrarily large (Theorem 15.4.22, again).

#### 15.4.2.4 Fast Convergence

We now turn, once again, to the issue of promoting fast convergence. Since we are using a nonsmooth merit function, we must be wary that the Maratos effect might interfere with the superlinear convergence of the underlying SQP method. As always, we shall inoculate our basic method by including a second-order correction when it is needed.

It is at this point that we notice a fundamental difference between the normal step calculated in Section 15.4.1.2 and that in 15.4.2.1. The former is obtained by backtracking from a step  $n_k^C$  to the linearized constraints—the step  $n_k^C$  is itself chosen if it lies within the (possibly shrunken) trust region  $\|n\| \leq \xi^N \Delta_k$ . For the current algorithm, there has been no requirement that the step satisfy the linearized constraints, even if such a step is possible. Recall that the only restriction we have placed on our step is that it should give a reduction in the linearized infeasibility comparable to that which would be obtained with a suitable Cauchy step. Such a weak requirement on the normal step is most unlikely to be enough to ensure fast overall convergence.

Ideally, we would like to find  $n_k$  to solve (15.4.113); we must bear in mind that, since there are likely many such minimizers, we also require that AA.1k hold. The best way to find such a point is to apply the truncated conjugate gradient method (Algorithm 7.5.1) to (15.4.113). Theorem 5.3.1 (p. 106) shows that all iterates generated by this method satisfy AA.1k. There are two possible outcomes: either the conjugate gradient path crosses the trust-region boundary, in which case

$$\|n_k\| = \xi^N \Delta_k \quad (15.4.151)$$

and there is no point satisfying

$$c(x_k) + A(x_k)n = 0 \text{ and } \|n\| \leq \xi^N \Delta_k,$$

or the method terminates at a point  $n_k$  for which

$$c(x_k) + A(x_k)n_k = 0 \text{ and } \|n_k\| \leq \xi^N \Delta_k. \quad (15.4.152)$$

In general, we shall require that  $n_k$  satisfies AA.1j and AA.1k and that either (15.4.151) or (15.4.152) holds. We shall only choose to consider a second-order correction if (15.4.152) is satisfied but  $\rho_k < \eta_1$ . This is a reasonable requirement, since only then is the constraint satisfied up to first order. Our second-order correction strategy is exactly as we described in Section 15.4.1.5. In particular, a full correction will be used when the standard SQP step has been taken; otherwise a partial correction will be used. The full algorithm is as follows.

**Algorithm 15.4.6: Algorithm 15.4.5 with a second-order correction**

The same as Algorithm 15.4.5, except that Step 1a is replaced by

**Step 1a: Normal step calculation.** Define a model  $m_k^N$  and compute a step  $n_k$  for which AA.1j and AA.1k hold and either

$$\|n_k\| = \xi^N \Delta_k \quad (15.4.153)$$

or

$$c(x_k) + A(x_k)n_k = 0 \text{ and } \|n_k\| \leq \xi^N \Delta_k \quad (15.4.154)$$

are satisfied.

and Steps 3 and 4 are replaced by

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$ ,  $c(x_k + s_k)$ , and  $\phi(x_k + s_k, \sigma_k)$  and define the ratio

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , then define  $x_{k+1} = x_k + s_k$  and possibly update  $H_{k+1}$ .

Otherwise, if (15.4.154) is satisfied,

**Step 3a: Second-order correction.** Choose  $H_k^{CS}$  and  $p_k$  so that  $K_k^{CS}$  is uniformly bounded and (15.4.82) is satisfied.

If  $s_k$  is a standard SQP step, choose  $g^{CS}$  to satisfy (15.4.81) and compute  $s_k^{CS} = s_k^{FC}$  from (15.4.79).

Otherwise, compute  $s_k^{CS} = s_k^{PC}$  from (15.4.85).

If

$$\|s_k^{CS}\| \leq \eta^{CS} \Delta_k, \quad (15.4.155)$$

**Step 3b: Attempt the second-order correction.** Redefine

$$\rho_k = \frac{\phi(x_k, \sigma_k) - \phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k)}{m(x_k, H_k, \sigma_k, 0) - m(x_k, H_k, \sigma_k, s_k)}.$$

If  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k + s_k^{\text{CS}}$  and possibly update  $H_{k+1}$ .

Otherwise, define  $x_{k+1} = x_k$ .

Otherwise, that is, if (15.4.155) fails, define  $x_{k+1} = x_k$ .

Otherwise, that is, if (15.4.154) fails, define  $x_{k+1} = x_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \quad x_{k+1} = x_k + s_k, \\ & \text{and (15.4.154) is satisfied,} \\ [\gamma_2 \Delta_k, \Delta_k] & \text{in all other cases when } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \quad (15.4.156)$$

Increment  $k$  by 1 and go to Step 1.

Although this algorithm appears complicated, its global convergence properties are easy to deduce.

**Theorem 15.4.23** Suppose that Algorithm 15.4.6 is replaced by Algorithm 15.4.2. Then Corollary 15.4.20, Theorem 15.4.22, and their supporting lemmas continue to hold.

**Proof.** As before, a careful examination of the proofs of Theorems 15.4.19 and 15.4.22 and their supporting lemmas shows that the introduction of a second-order correction does not alter the conclusion of these results. The only places where we need to be careful are in the proof of Theorem 15.4.19, where (15.4.138) now requires the additional bound (15.4.155) when second-order corrections are used, and in that of Theorem 6.4.6 (p. 137), where (6.4.22) (p. 138) requires the same bound.  $\square$

Turning to the asymptotic convergence behaviour, we have the following result.

**Theorem 15.4.24** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.6 have a limit point  $(x_*, y_*)$  at which AO.1b and AO.3 hold. Suppose further that AW.1b, AW.1c, AW.1f, AM.4j, AA.12, AA.13, and AA.14 hold. Finally suppose that  $H_k = H(x_k, y_k)$ , that  $\sigma_k^B = \|y_k\| + \tau$  for some small  $\tau > 0$ , and that the standard SQP step is taken whenever possible for all  $(x_k, y_k)$  close to  $(x_*, y_*)$ . Then  $\{x_k\}$  converges to  $x_*$  Q-superlinearly. If  $\|y_k - y_*\| = O(\|x_k - x_*\|)$ , the rate is Q-quadratic.

**Proof.** The proof is very similar to that of Theorem 15.4.15. The assumptions are sufficient for Theorem 15.4.23 to ensure that the penalty  $\sigma_k$  parameter is ultimately fixed at some  $\sigma_{\max} > \|y_*\|_2$ , and that the complete sequence of iterates  $\{x_k\}$  ultimately converges to a strict, isolated local minimizer  $x_*$  of  $\phi(x, \sigma_{\max})$ . We proceed to show that there is a neighbourhood of  $x_*$  in which all iterations are successful.

Suppose first that

$$\|n_k\|_2 \geq \frac{\chi\xi^N}{\xi} \|s_k\|_2 \quad (15.4.157)$$

for some  $0 < \xi < 1$ . Step 2 of the algorithm ensures that (15.4.7) is satisfied, while this, (15.4.141), (15.4.149), and (15.4.157) give that

$$\delta m_k \geq \nu \sigma_k \delta m_k^N \geq \nu \sigma_k \kappa_{\text{bon}} \|n_k\|_2 \geq \kappa_6 \|s_k\|_2,$$

where

$$\kappa_6 \stackrel{\text{def}}{=} \frac{\nu \sigma_{\max} \kappa_{\text{bon}} \chi \xi^N}{\xi}.$$

Thus, exactly as in the proof of Theorem 15.4.15, we deduce that the iteration will be very successful for all  $\Delta_k$  smaller than the bound (15.4.97), and hence the radius will not decrease in this case.

So now suppose that

$$\|n_k\|_2 < \frac{\chi\xi^N}{\xi} \|s_k\|_2 \quad (15.4.158)$$

but that  $x_k + s_k$  is unsuccessful. Then, it follows from (15.4.34) and (15.4.158) that

$$\|n_k\|_2 < \xi^N \Delta_k$$

and thus that (15.4.153) does not hold. Hence, by design, (15.4.154) holds and a second-order correction will be computed. Exactly the same reasoning as in the proof of Theorem 15.4.15 implies that the iteration will be very successful for all sufficiently small  $\Delta_k$ , and the radius will not decrease in this case. The final case is when (15.4.98) is satisfied and  $x_k + s_k$  is successful. In this case, (15.4.95) requires that the radius will be either unaltered or perhaps increased.

We have thus seen that all iterations will ultimately be successful, and the radius will be bounded away from zero. Since  $s_k$  converges to zero, the trust region must ultimately be inactive, and (15.4.154) will hold. If  $s_k$  is accepted, the Q-superlinear and possible Q-quadratic convergence follow from Theorem 15.2.1. If this step is rejected, the fully corrected step must be accepted, and the Q-superlinear and possible Q-quadratic convergence in this case follow from (15.3.45).  $\square$

#### 15.4.2.5 Convergence to Second-Order Critical Points

The reader will not be surprised to find that the results we derived in Section 15.4.1.6 about convergence to second-order critical points apply equally to the algorithm we have been examining in this section. As before, we need to ensure that the tangential

step is chosen to satisfy the additional condition AA.2d when there is negative curvature in the null-space of the gradients of the constraints, and that we may need to resort to a (partial) second-order correction to escape from regions where the curvature of the constraints is dominant because of the BCCSS effect. The following algorithm is appropriate.

**Algorithm 15.4.7: A second-order variant of Algorithm 15.4.6**

The same as Algorithm 15.4.6, except that Step 1b is replaced by

**Step 1b: Tangential step calculation.** Define a model  $m_k^T$  and compute a step  $t_k$  for which  $\|t_k\| \leq \xi^T \Delta_k$  is satisfied and AA.1i and AA.2d hold.

As before, the extra step condition AA.2d does not change the conclusions of Theorem 15.4.23, and in particular any limit point is at least a first-order critical point. We may easily deduce the following second-order convergence result.

**Theorem 15.4.25** Suppose that the iterates  $\{x_k\}$  and related Lagrange multiplier estimates  $\{y_k\}$  generated by Algorithm 15.4.7 have a finite number of limit points  $(x_*^i, y_*^i)$ ,  $1 \leq i \leq \ell$ , at which AO.1b holds. Suppose further that AW.1b, AW.1c, AW.1f, AM.4j, AA.12, AA.13, and AA.14 hold. Finally suppose that  $H_k = H(x_k, y_k)$ . Then at least one of the  $x_*^i$  is a strong second-order critical point for the problem (15.1.2).

**Proof.** The proof is essentially the same as that for Theorem 15.4.16. We suppose, as before, that

$$\tau_*^i = \lambda_{\min}[N^T(x_*^i)H(x_*^i, y_*^i)N(x_*^i)] < 0 \quad (15.4.159)$$

for all  $1 \leq i \leq \ell$ , and deduce that  $(x_k, y_k) \in \mathcal{N}_*$  for  $k$  sufficiently large, where

$$2\lambda_{\min}[N^T(x)H(x, y)N(x)] \leq \tau_* \stackrel{\text{def}}{=} \min_{1 \leq i \leq \ell} \tau_*^i < 0$$

for all  $(x, y) \in \mathcal{N}_*$ .

If  $\|n_k\|_2 > \chi \xi^N \Delta_k$  for some  $0 < \chi < 1$ , it follows exactly as in the proof of Theorem 15.4.16 that the iteration will be very successful, and

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k) \geq \eta_2 \kappa_7 \Delta_k^2 \quad (15.4.160)$$

for all  $\Delta_k$  sufficiently small. If, on the other hand,  $\|n_k\|_2 \leq \chi \xi^N \Delta_k$ , (15.4.153) does not hold, and thus by design (15.4.154) holds. Since  $(x_k, y_k) \in \mathcal{N}_*$ ,  $\tau_k \leq \frac{1}{2}\tau_* < 0$ , (15.4.150), and AA.2d together imply that

$$\delta m_k \geq \kappa_8 \Delta_k^2, \quad \text{where } \kappa_8 = -\frac{1}{2}\kappa_{\text{btm}}\kappa_{\text{sod}}\tau_* > 0.$$

Thus, if the step  $s_k$  is successful,

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k, \sigma_k) \geq \eta_1 \kappa_s \Delta_k^2, \quad (15.4.161)$$

and (15.4.156) implies that the radius does not decrease. If the step is unsuccessful, the partial second-order correction  $s_k^{\text{CS}} = s_k^{\text{PC}}$  will be contemplated since (15.4.154) holds. Exactly as in the proof of Theorem 15.4.16, it then follows that the second-order correction will be permitted, it will turn out to be very successful, the radius will not decrease, and

$$\phi(x_k, \sigma_k) - \phi(x_k + s_k + s_k^{\text{CS}}, \sigma_k) \geq \eta_2 \delta m_k \geq \eta_2 \kappa_s \Delta_k^2 \quad (15.4.162)$$

for all sufficiently small  $\Delta_k$ .

Thus, since all iterates are ultimately within  $\mathcal{N}_*$ , the trust-region radii satisfy

$$\Delta_k \geq \Delta_{\max} \quad (15.4.163)$$

for some  $\Delta_{\max} > 0$  and all  $k \geq 0$ . Hence there must be an infinite number of successful iterations, and on each of these (15.4.160), (15.4.161), (15.4.162), and (15.4.163) show that

$$\phi(x_k, \sigma_{\max}) - \phi(x_{k+1}, \sigma_{\max}) \geq \kappa_9 \Delta_{\max}^2$$

for some  $\kappa_9 > 0$ . But this is impossible since AW.1d implies that  $\phi(x, \sigma_{\max})$  is bounded from below. Hence our assumption (15.4.159) must be false,

$$\lambda_{\min}[N^T(x_*^i)H(x_*^i, y_*^i)N(x_*^i)] \geq 0$$

for some  $1 \leq i \leq \ell$ , and thus  $x_*^i$  is a strong second-order critical point for (15.1.2).  $\square$

## Notes and References for Subsection 15.4.2

The basic method we consider here was first suggested by Omojokun (1989) in his doctoral thesis, supervised by R. H. Byrd. This aptly named Byrd–Omojokun approach forms the basis for the developments by Biegler, Nocedal, and Schmid (1995), El-Alem (1995a, 1999), Byrd, Gilbert, and Nocedal (1996), Bielschowsky and Gomes (1998), Liu and Yuan (1998), Lalee, Nocedal, and Plantenga (1998), and Byrd, Hribar, and Nocedal (2000), from which the ETR and NITRO software packages have evolved. See also Gomes, Maciel, and Martínez (1999).

The analysis in this section is based heavily on that of Byrd, Gilbert, and Nocedal (1996). Their method is actually for problems involving both equality and inequality constraints. Any inequality is converted into an equation, and the resulting slack variable treated by a suitable barrier function—the minimization of this barrier function provides an additional outer iteration. The different scaling required to treat the original and slack variables adds further complications, as we have already seen in Chapter 13. Difficulties also arise because the slack variables are explicitly bounded away from zero in both the normal and tangential

subproblems; these extra constraints have the slightly unfortunate side-effect that the solution of the subproblem can no longer be computed directly using the techniques developed in Chapter 7. See Section 13.3 for more details and Section 13.2 for alternatives. A variety of useful details and successful heuristics were given by Lalee, Nocedal, and Plantenga (1998). Omojokun (1989) analysed the local convergence of such a method and showed that the Maratos effect is avoided by a suitable second-order correction and that Q-quadratic convergence may be obtained using suitable approximations  $H_k$ .

Dennis, El-Alem, and Maciel (1997) and El-Alem (1996b) considered an alternative to the Byrd–Omojokun method using the augmented Lagrangian function, rather than the  $\ell_2$  exact penalty function, as a merit function. Their analysis is quite similar to that given in this section, and they show that, under broadly identical assumptions that include AA.14, the generated iterates may be made arbitrarily close to a first-order critical point. Instead of our assumption AA.1k, they prefer simply to require that the normal step satisfy  $\|n_k\| \leq \kappa_4 \|c(x_k)\|$  for all  $k$  and some constant  $\kappa_4 > 0$ . Such a condition, or indeed a requirement that  $\|n_k^N\| \leq \kappa_5 \|c(x_k)\|$  for some other constant  $\kappa_5 > 0$ , might equally be applied in our analysis to deduce that (15.4.147) holds. Although Dennis, El-Alem, and Maciel (1997) show that the penalty parameter for the augmented Lagrangian remains bounded in a neighbourhood of the desired critical point, there is nothing to prevent the parameter from approaching infinity if this neighbourhood is required to shrink to zero. This is hardly surprising since no attempt is made to compute useful Lagrange multipliers—in which case the use of the merit function may more correctly be interpreted as if it were a shifted quadratic penalty function for which the parameter must approach infinity (see Section 14.3)—and we can only conjecture that this defect will vanish if first-order or least-squares multiplier estimates are used.

A number of extensions of this analysis are possible. By relaxing AA.14 and strengthening AA.1k, El-Alem (1995a, 1999) shows that the iterates may still be made arbitrarily close to a first-order critical point (at best) or a critical point for the infeasibility (at worst). Dennis and Vicente (1997) extend Dennis, El-Alem, and Maciel’s (1997) method to show convergence to a second-order critical point so long as the step obtained for the tangential-step subproblem gives a model reduction within a fixed fraction of the best possible reduction.

### 15.4.3 Celis–Dennis–Tapia-like Approaches

A third way of dealing with the possibility that the linearized constraints (15.2.9b) and the trust region (15.2.9c) have no common feasible point is to replace the former by

$$\|A(x)s + c(x)\| \leq \theta, \quad (15.4.164)$$

where  $\theta$  is chosen so that the intersection of (15.4.164) and the trust region is nonempty. Clearly, since we wish to reduce the infeasibility, we should insist at the very least that

$$\min_{\|s\| \leq \Delta} \|A(x)s + c(x)\| \leq \theta \leq \|c(x)\|, \quad (15.4.165)$$

while another possibility is to require that

$$\min_{\|s\| \leq \xi_1 \Delta} \|A(x)s + c(x)\| \leq \theta \leq \min_{\|s\| \leq \xi_2 \Delta} \|A(x)s + c(x)\|, \quad (15.4.166)$$

where  $0 < \xi_2 \leq \xi_1 < 1$ . Solving problems of the form

$$\min_{\|s\| \leq \xi\Delta} \|A(x)s + c(x)\|$$

for some  $0 < \xi \leq 1$ , which are needed to ensure that  $\theta$  satisfies (15.4.165) or (15.4.166), may be expensive. A cheaper possibility is to find *any* step  $n$  that lies within the trust region but that also significantly reduces  $\|A(x)n+c(x)\|$ . The most popular choice is, of course, the Cauchy step (15.4.116), but any step that further decreases  $\|A(x)n+c(x)\|$  is also possible.

Although the computation of a suitable value  $n$  to reduce the infeasibility is reminiscent of the composite step methods considered,  $n_k$  is actually only used to find

$$\theta_k = \|A(x_k)n_k + c(x_k)\| \leq \|c(x_k)\|, \quad (15.4.167)$$

where the inequality in (15.4.167) is strict unless  $c(x_k) = 0$ . The overall step is computed as an approximate solution to the problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\langle s, H_k s \rangle + \langle \nabla_x \ell(x_k, y_k), s \rangle \quad (15.4.168a)$$

$$\text{subject to} \quad \|A(x_k)s + c(x_k)\| \leq \theta_k \quad (15.4.168b)$$

$$\text{and} \quad \|s\| \leq \Delta_k \quad (15.4.168c)$$

for some appropriate Lagrange multiplier estimates  $y_k$  and approximation,  $H_k$ , to the Hessian of the Lagrangian. Notice that by considering the whole feasible region (15.4.168b) and (15.4.168c) rather than successive normal and tangential components, there is a clear potential for greater reductions in the objective function (15.4.168a) than with the previous two approaches. Unfortunately, this advantage may also be regarded as its Achilles' heel.

The main disadvantage of these approaches is apparent if one considers (15.4.168). If polyhedral norms are used, this subproblem reduces to a (possibly nonconvex) inequality-constrained quadratic program that may prove rather expensive to solve. On the other hand, if we choose the  $\ell_2$  norm, the subproblem involves *two* quadratic constraints. Thus it is unclear if or how the special techniques we developed in Chapter 7 for the simpler subproblem involving a single quadratic constraint may or can be applied. In particular, it is far from evident how to compute the true model minimizer, nor is it obvious how to derive a useful approximation. Indeed, given that global and local convergence theories matching those of the other methods we have considered in this chapter can be developed, we can only surmise that the lack of any reported implementation based on the approach taken in this section may be attributed to this disadvantage.

In view of this drawback, we do not intend to give a detailed global convergence analysis of methods based on (15.4.168), but refer the interested reader to the works cited in the notes at the end of this section for further details. Nevertheless, we shall still summarize the basic means by which we can embed the subproblem (15.4.168) within a globally convergent method.

As usual, having chosen a means of generating a step, we need to introduce an appropriate merit function. We must also show that the generated step significantly decreases this merit function at any noncritical point for all sufficiently small trust-region radii. Although it is most likely possible to develop globally convergent methods based on a nondifferentiable penalty function  $f(x) + \sigma\|c(x)\|$ , all methods we are aware of prefer the augmented Lagrangian function (15.3.1) or the smooth exact penalty functions (15.3.50). The reason is quite simply that there appears to be an inherent consistency between these functions and the subproblem (15.4.168) when the  $\ell_2$  norm is used for (15.4.168b) and (15.4.168c).

To see this, note that the first-order criticality conditions (3.2.5)–(3.2.8) (pp. 40, 41) for the equivalent subproblem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}\langle s, H_k s \rangle + \langle \nabla_x \ell(x_k, y_k), s \rangle \quad (15.4.169a)$$

$$\text{subject to} \quad \theta_k^2 - \|A(x_k)s + c(x_k)\|_2^2 \geq 0 \quad (15.4.169b)$$

$$\text{and} \quad \Delta_k^2 - \|s\|_2^2 \geq 0 \quad (15.4.169c)$$

are that the solution  $s_k$  of (15.4.168) satisfy

$$\left( H_k + \sigma_k A^T(x_k)A(x_k) + \mu_k I \right) s_k = - \left( g(x_k) - A^T(x_k)(y_k - \sigma_k c(x_k)) \right)$$

for some Lagrange multipliers  $\sigma_k, \mu_k \geq 0$  for which

$$\sigma_k(\|A(x_k)s + c(x_k)\|_2 - \theta_k) = 0 \quad \text{and} \quad \mu_k(\|s\|_2 - \Delta_k) = 0.$$

Now comparing this equation with the Newton equations (14.3.17) (p. 588),

$$\left( \nabla_{xx} \ell(x_k, y_k^F) + \frac{1}{\mu_k} A^T(x_k)A(x_k) \right) s_k = - \left( g - A^T(x_k)y_k^F \right),$$

where  $y_k^F = y_k - c(x_k)/\mu_k$  for the augmented Lagrangian function, we see that the two are the same if  $\sigma_k = 1/\mu_k$  and  $H_k = \nabla_{xx} \ell(x_k, y_k^F)$  whenever the trust-region bound is inactive. That is,  $s_k$  is the Newton correction for the augmented Lagrangian function for an appropriate value of the penalty parameter so long as the trust region is inactive. It is straightforward to show that this  $s_k$  is sure to be a descent direction for the merit function whenever  $\mu_k \leq 1/\sigma_k$  regardless of the trust region, so long as  $H_k + \sigma_k A^T(x_k)A(x_k)$  is positive definite.

We thus consider the merit function

$$\Phi(x, y, \sigma) = f(x) + \langle c(x), y \rangle + \frac{1}{2}\sigma\|c(x)\|_2^2$$

for some appropriate penalty parameter  $\sigma$  as a function of both  $x$  and  $y$ . Changes  $s_k$  to  $x_k$  will be computed as an approximate solution of (15.4.168), while a change  $s_k^y$  to  $y_k$  is chosen by other appropriate means, including simply setting  $s_k^y = 0$ . As usual we shall accept or reject candidate steps by comparing the difference  $\Phi(x_k, y_k, \sigma_k) - \Phi(x_k + s_k, y_k + s_k^y, \sigma_k)$  in the merit function to an appropriate model of this difference. The most convenient model of  $\Phi(x + s, y + s^y, \sigma)$  is the Taylor approximation

$$\begin{aligned} m(x, y, H, \sigma, s, s^y) &= \Phi(x, y, \sigma) + \langle \nabla_x \ell(x, y), s \rangle + \frac{1}{2}\langle s, H s \rangle \\ &\quad - \langle c(x) + A(x)s, s^y \rangle + \frac{1}{2}\sigma\|c(x) + A(x)s\|_2^2, \end{aligned}$$

although the alternative

$$\begin{aligned} m(x, y, H, \sigma, s, s^y) &= \Phi(x, y, \sigma) + \langle \nabla_x \ell(x, y(x)), s \rangle + \frac{1}{2} \langle s, H s^N \rangle \\ &\quad - \langle y(x + s) - y(x), c(x) + \frac{1}{2} A(x)s \rangle + \frac{1}{2} \sigma \|c(x) + A(x)s\|_2^2, \end{aligned} \quad (15.4.170)$$

where  $s^N$  is the orthogonal projection of  $s$  into the null-space of  $A(x)$ , has been proposed as a suitable alternative when the merit function (15.3.50) is used.

Of course, there is no automatic guarantee that the model change

$$\delta m = m(x, y, H, \sigma, 0, 0) - m(x, y, H, \sigma, s, s^y)$$

will be positive, and we may be forced to increase  $\sigma$  to ensure that this is so. Rather, as we did in Section 15.4.1.1, we shall require that

$$\delta m \geq \frac{1}{2} \nu \sigma \left[ \|c(x)\|_2^2 - \|c(x) + A(x)s\|_2^2 \right], \quad (15.4.171)$$

where we note that the right-hand-side of (15.4.171) is strictly positive when  $c(x) \neq 0$  because of our choice of  $\theta$  in (15.4.167). If we let

$$\begin{aligned} \delta m^N &= \|c(x)\|_2^2 - \|c(x) + A(x)s\|_2^2 \quad \text{and} \\ \delta m^F &= \langle \nabla_x \ell(x, y), s \rangle + \frac{1}{2} \langle s, H s \rangle - \langle c(x) + A(x)s, s^y \rangle, \end{aligned}$$

then, just as we did in Section 15.4.1.1, we may replace the current  $\sigma$  by  $\max[\sigma^C, \tau_1 \sigma, \sigma + \tau_2]$  whenever  $\sigma < \sigma^C$ . Here

$$\sigma^C = \max \left[ \sigma^B, -\frac{2\delta m^F}{(1-\nu)\delta m^N} \right]$$

for some appropriate  $\sigma^B$ ,  $\tau_1 \geq 1$ , and  $\tau_2 \geq 0$ , for which the product  $(\tau_1 - 1)\tau_2 > 0$ . This ensures that each change in  $\sigma$  is by at least a nontrivial amount,  $\tau_2$ . When  $c(x) = 0$ , there is no need to change  $\sigma$ , since then  $\theta = 0$  and hence  $c(x) + A(x)s = 0$ , and any reduction in (15.4.169a) will also reduce  $m(x, y, H, \sigma, s, s^y)$ . We might also replace  $\delta m^F$  by

$$\delta m^F = \langle \nabla_x \ell(x, y(x)), s \rangle + \frac{1}{2} \langle s, H s^N \rangle - \langle y(x + s) - y(x), c(x) + \frac{1}{2} A(x)s \rangle$$

in the above if the exact augmented Lagrangian merit function (15.3.50) is preferred.

We now have all the ingredients necessary to describe a generic algorithm, Algorithm 15.4.8.

#### Algorithm 15.4.8: An infeasibility-reducing SQP algorithm

**Step 0: Initialization.** An initial point  $x_0$ , Lagrange multiplier estimate  $y_0$ , trust-region radius  $\Delta_0 > 0$ , and penalty parameter  $\sigma_{-1} > 0$  are given. The constants  $\eta_1, \eta_2, \gamma_1, \gamma_2, \nu, \tau_1$ , and  $\tau_2$  are also given and satisfy the conditions

$$\begin{aligned} 0 < \eta_1 \leq \eta_2 < 1, \quad 0 < \gamma_1 \leq \gamma_2 < 1, \\ 0 < \nu < 1, \quad \tau_1 \geq 1, \tau_2 \geq 0 \quad \text{and} \quad (\tau_1 - 1)\tau_2 > 0. \end{aligned}$$

Compute  $f(x_0)$  and  $c(x_0)$ , and set  $k = 0$ .

**Step 1: Step calculation.** Compute the steps  $s_k$  and  $s_k^y$  as follows:

**Step 1a: Reduce the infeasibility.** If  $c(x_k) \neq 0$ , compute a step  $n_k$  to reduce  $\|c(x_k) + A(x_k)n\|_2$  while satisfying  $\|n\| \leq \Delta_k$ , and set  $\theta_k = \|c(x_k) + A(x_k)n_k\|_2$ . Otherwise, set  $\theta_k = 0$ .

**Step 1b: Compute the step  $s_k$ .** Compute an approximate solution  $s_k$  to (15.4.168).

**Step 1c: Compute the step  $s_k^y$ .** Compute an appropriate correction  $s_k^y$ .

**Step 2: Increase the penalty parameter if necessary.** Compute the changes  $\delta m_k^N$  and  $\delta m_k^F$ . Find

$$\sigma_k^C = \max \left[ \sigma_k^B, -\frac{2\delta m_k^F}{(1-\nu)\delta m_k^N} \right]$$

with some suitable  $\sigma_k^B$ ; set

$$\sigma_k = \begin{cases} \max[\sigma_k^C, \tau_1 \sigma_{k-1}, \sigma_{k-1} + \tau_2] & \text{if } \sigma_{k-1} < \sigma_k^C, \\ \sigma_{k-1} & \text{otherwise;} \end{cases}$$

and compute the overall model values

$$m(x_k, y_k, H_k, \sigma_k, 0, 0) \equiv \phi(x_k, y_k, \sigma_k) \quad \text{and} \quad m(x_k, y_k, H_k, \sigma_k, s_k, s_k^y).$$

**Step 3: Acceptance of the trial point.** Compute  $f(x_k + s_k)$ ,  $c(x_k + s_k)$ , and  $\phi(x_k + s_k, y_k + s_k^y, \sigma_k)$  and define the ratio

$$\rho_k = \frac{\phi(x_k, y_k, \sigma_k) - \phi(x_k + s_k, y_k + s_k^y, \sigma_k)}{m(x_k, y_k, H_k, \sigma_k, 0, 0) - m(x_k, y_k, H_k, \sigma_k, s_k, s_k^y)}.$$

If  $\rho_k \geq \eta_1$ , define  $x_{k+1} = x_k + s_k$  and  $y_{k+1} = y_k + s_k^y$ , and possibly update  $H_{k+1}$ .

Otherwise define  $x_{k+1} = x_k$  and  $y_{k+1} = y_k$ .

**Step 4: Trust-region radius update.** Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

Increment  $k$  by 1 and go to Step 1.

Reasonable values for the parameters in Algorithm 15.4.8 are exactly as in (15.4.27). We have deliberately not given details on how one would choose the all-important steps  $n_k$ ,  $s_k$ , and  $s_k^y$ , and we reiterate our principal concern that the determination of a useful  $s_k$  is far from trivial.

### Notes and References for Subsection 15.4.3

The choice (15.4.167), using the Cauchy point (15.4.116) for  $n$ , is due to Celis (1985) and Celis, Dennis, and Tapia (1985), while (15.4.166) was given by Powell and Yuan (1990). Variations on the former were given by El-Alem (1988, 1991, 1996a), Williamson (1990), and Ke and Han (1995b), and a general convergence theory was given by Dennis, El-Alem, and Maciel (1997). A similar device was proposed by Burke and Han (1989) to handle inconsistent constraints in the basic subproblem (15.2.4). Burke (1992) studied methods of this sort in a very general setting and allowed for the possibility that the original problem may be infeasible by showing convergence to a “nearest” infeasible KKT point.

The global convergence analyses of Powell and Yuan (1990), El-Alem (1991), and Dennis, El-Alem, and Maciel (1997) all came to a broadly similar conclusion, namely, that if the columns of  $N(x)$  provide an acceptable basis for the null-space of  $A(x)$  (in the sense of AA.12) and other standard assumptions are satisfied, the criticality measures  $\|c(x_k)\| + \|\nabla_x \ell(x_k, y_k)\|$  or  $\|c(x_k)\| + \|N^T(x_k)g(x_k)\|$  may be reduced below a predefined (arbitrarily small) positive tolerance  $\epsilon$ , and that the penalty parameter is bounded by some function of  $1/\epsilon$ . This does not, of course, imply that the penalty parameter stays bounded if  $\epsilon$  converges to zero, nor necessarily that any limit point is a first-order critical point. Powell and Yuan (1990) also showed that if their method, which uses the exact augmented Lagrangian function (15.3.50) together with the model (15.4.170), has a single limit point, the iterates converge at least Q-superlinearly under standard second-order assumptions. Notice that it is this asymptotic convergence analysis that requires that (15.4.170) involve the slightly strange contributions  $s^N$  and  $c(x) + \frac{1}{2}A(x)s$ . The use of the smooth exact penalty function is supposedly to avoid the Maratos effect, but since we appear to have traded this for the possibility of an arbitrarily large penalty parameter and the subsequent ill-conditioning this entails (see Section 14.3), it is debatable whether the trade is actually worthwhile.

Originally, the  $\ell_2$  norm was suggested for (15.4.168), although El-Alem and Tapia (1995) indicated that there may be some advantages in using a polyhedral norm, since the subproblem may then simply be written as a quadratic program. When the  $\ell_2$  norm is used, Williamson (1990), Yuan (1990, 1991, 1998c), Chen (1996), and Chen and Yuan (1999) categorized the solutions of and give algorithms for the subproblem. Ecker and Niemi (1975), Phan huy Hao (1982), Yuan (1990), Mehrotra and Sun (1991), Zhang (1992), Heinkenschloss (1994), and Martínez and Santos (1995) simplified such analyses in the special case where  $H$  is positive semidefinite. For some progress on the general case, see Fu, Luo, and Ye (1996). Recent work by Moré (1993), Stern and Wolkowicz (1995), Ben-Tal and Teboulle (1996), Peng and Yuan (1997), Heinkenschloss (1998), Kruk and Wolkowicz (1998), and Le Thi (2000) has generalized the above to cases where (15.2.9c) is replaced by the condition  $\Delta_1 \leq \langle s, Cs \rangle \leq \Delta_2$  and where  $C$  may be indefinite, or where more than two quadratic constraints are present.

### 15.4.4 Inequality Constraints

The methods we have discussed in this section have all been concerned with the equality problem (15.1.2). We have felt comfortable with this for two reasons. Firstly, the methods were easier to motivate and develop for the equality case. Secondly, we have already seen other tools for handling inequality constraints in Chapters 12 and 13. In this section, however, we shall return to how we might deal with problems like (15.1.1) that involve inequality constraints (or a mixture of both).

There are two basic approaches. The first is to extend the model problem to include inequalities. Typically, this means that we would ideally like to solve an inequality-constrained model problem of the form

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \langle s, H_k s \rangle + \langle g(x_k), s \rangle \quad (15.4.172a)$$

$$\text{subject to} \quad A_{\mathcal{E}}(x_k)s + c_{\mathcal{E}}(x_k) = 0, \quad A_{\mathcal{I}}(x_k)s + c_{\mathcal{I}}(x_k) \geq 0, \quad (15.4.172b)$$

$$\text{and} \quad \|s\| \leq \Delta_k \quad (15.4.172c)$$

(see Section 15.2.2). Since this problem may not have a solution when  $\Delta_k$  is small and the current iterate is infeasible, we may instead have to be content with a step that moves us towards a solution to the model problem. All the methods we have considered in Section 15.4 have achieved this by decomposing the step as

$$s_k = n_k + t_k,$$

where the normal step  $n_k$  is chosen to reduce the (linearized) infeasibility, and the tangential step  $t_k$  is then determined to reduce the model without worsening the infeasibility attained during the normal step. For the general problem, much the same approach is valid.

There are obvious variants of all of the three main approaches we have discussed. Consider first the normal step. To extend the Vardi-like methods we described in Section 15.4.1, we need to compute a trial step  $n_k^C$  that satisfies the linear constraints (15.4.172b) and that is not significantly longer than the projection onto the linearized feasible region. We then take a step  $\alpha_k$  in this direction as far, or almost as far, as we can within the trust region—the requirement AA.1h is ideal—and set  $n_k = \alpha_k n_k^C$ . For the Byrd–Omojokun-like approaches of Section 15.4.2, the normal step should be calculated by finding  $n_k$  to approximately

$$\underset{n \in \mathbb{R}^n}{\text{minimize}} \quad \left\| (A(x_k)n + c(x_k))^{\mathcal{I}-} \right\| \quad \text{subject to} \quad \|n\| \leq \xi^N \Delta_k \quad (15.4.173)$$

for some  $0 < \xi^N < 1$ , essentially as we did in (15.4.113). Note that (15.4.173) can be reformulated as a convex quadratic program, and thus, in principle, there are effective methods for (approximately) solving it. Finally, to extend the Celis–Dennis–Tapia-like approaches of Section 15.4.3, we merely need the normal step to give us at least as much reduction in  $\|(A(x_k)n + c(x_k))^{\mathcal{I}-}\|$  as a step to the generalized Cauchy point for this problem.

Turning to the tangential step, extensions to both the Vardi-like and Byrd–Omojokun-like approaches require that the step approximately

$$\begin{aligned} \underset{t \in \mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2} \langle t, H_k t \rangle + \langle g(x_k) + H_k n_k, t \rangle \\ \text{subject to} \quad & A_{\mathcal{E}}(x_k) t = 0, \quad A_{\mathcal{I}}(x_k) t \geq 0, \\ \text{and} \quad & \|t\| \leq \xi^T \Delta_k \end{aligned}$$

for some  $0 < \xi^T \leq 1$ . Notice that the linearized infeasibility is made no worse, and attention turns instead to reducing the model value. In theory, all that is required is that the reduction in the model at  $t_k$  be a positive fraction of that attainable at a generalized Cauchy point, such as we considered in Section 12.2.1. For Celis–Dennis–Tapia-like approaches, the tangential step must be calculated to approximately

$$\begin{aligned} \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2} \langle s, H_k s \rangle + \langle \nabla_x \ell(x_k, y_k), s \rangle \\ \text{subject to} \quad & \left\| (A(x_k)s + c(x_k))^{\mathcal{I}^-} \right\| \leq \theta_k \\ \text{and} \quad & \|s\| \leq \Delta_k, \end{aligned} \tag{15.4.174}$$

where  $\theta_k = \|(A(x_k)n_k + c(x_k))^{\mathcal{I}^-}\|$ . As before, this approach is less attractive in practice than its predecessors, as effective methods for approximately minimizing (15.4.174) are not known.

Other details extend in an obvious way. In particular, the same merit functions as before are appropriate, provided we replace all mention of  $c(x)$  by  $c(x)^{\mathcal{I}^-}$ . We leave a full description of detailed algorithms, and the consequent convergence theory, to the reader’s imagination.

The second way of moving from the equality-constrained to the general problem is to handle inequalities as we did in Chapter 13. That is, we embed the general problem (15.1.1) within a sequence of equality-constrained problems of the form

$$\begin{aligned} \text{minimize} \quad & f(x) + b_{\mathcal{I}}(x, \mu_k) \\ \text{subject to} \quad & c_i(x) = 0 \text{ for } i \in \mathcal{E}, \end{aligned}$$

where  $b_{\mathcal{I}}(x, \mu)$  is one of the barrier functions considered in Section 13.9 and  $\{\mu_k\}$  is a sequence of barrier parameters that converge to zero from above. For this class of methods, we insist that we start from a strictly feasible point for the inequality constraints, that is, that  $c_{\mathcal{I}}(x) > 0$  and that all subsequent iterates remain strictly feasible for these constraints.

The simplest framework is to wrap the algorithms of Sections 15.4.1–15.4.3 inside a primal or primal-dual barrier iteration like Algorithms 13.4.1 or 13.6.2, modelling the barrier function  $f(x) + b_{\mathcal{I}}(x, \mu_k)$  as we do in Section 13.9. However, since such quadratic models have little influence in dissuading the iterates from violating one or more of our inequality constraints, we can adopt either (or both) of the strategies proposed in Chapter 13 with this aim: either adjust the shape of the trust region to keep the iterates feasible, or add explicit extra constraints to the trust-region subproblems to do this.

The main difficulties when there are nonlinear inequality constraints present are that measuring the distance to feasibility may be hard, and any additional constraints imposed on the trust-region subproblem may be nonlinear. For these reasons, inequality constraints are often converted to equations by introducing slack variables. To be specific, we may replace  $c_{\mathcal{I}}(x) \geq 0$  by the equivalent

$$c_{\mathcal{I}}(x) - x^s = 0 \text{ and } x^s \geq 0,$$

where  $x^s$  are known as *slack variables*. We thus have replaced nonlinear inequality constraints by nonlinear equalities and have introduced corresponding extra variables that are required to be nonnegative. The advantage of doing so is that we believe that the methods given throughout Section 15.4 are well able to deal with nonlinear equality constraints, while the barrier/interior-point, Coleman and Li, and affine-scaling methods discussed in Chapter 13 are especially suited to linear, and particularly simple bound, constraints.

There are still some disadvantages of adding slack variables. Firstly, we have most definitely increased the dimension of the problem. To counter this, it is important to realize that the dominant cost of most algorithms (at least when function values are inexpensive) tends to be that for the linear algebra. In practice, significant algebraic savings may be made by recognizing that slack variables only occur linearly in the problem reformulation, and each slack variable is associated with a single constraint. The second disadvantage is that a suitable scaling of the slack variables is often difficult to find—in practice, it is more usual to pick the slacks so that

$$c_{\mathcal{I}}(x) - Dx^s = 0 \text{ and } x^s \geq 0,$$

where the diagonal matrix  $D$  is supposed to reflect “typical” values of  $c_{\mathcal{I}}(x)$ , but the very fact that  $c$  is nonlinear indicates that a uniformly good  $D$  may be hard to determine. This has further repercussions for trust-region methods since it is usual to scale the trust-region norm to account for different scalings of the variables. We also note that in practice the trust-region scaling needs to reflect the interaction between the nonlinear constraints and the simple bounds. At every iteration, a preconditioner of the form (13.10.18) (p. 547), in which  $A$  represents the linear(ized) constraints, is consistent with the trust-region (semi)norm (13.10.18) (that is, the dual of (13.10.17)) and is recommended.

We conclude this short section on inequality constraints with the remark that blending good methods for coping with equality constraints with good ones for dealing with inequalities is an extremely active area of research. For this reason, we shall say no more here, but await further developments, and particularly comparisons of the numerous possibilities, with interest.

### Notes and References for Subsection 15.4.4

We have already mentioned that Byrd, Gilbert, and Nocedal (1996) have extended the Byrd–Omojokun method to handle inequalities by converting these to equations using slack variables, and dealing with the latter using primal and primal-dual interior-point techniques.

A similar approach is taken by Plantenga (1999). The resulting software packages NITRO (see Byrd, Hribar, and Nocedal, 2000) and BECTR (see Plantenga, 1999) are, together with Gomes, Maciel, and Martínez (1999), the first we are aware of that treat inequality as well as equality constraints in a composite-step framework.

Some problems, most especially those that arise from discretizations of optimal control problems, naturally occur in the form

$$\underset{\substack{x^y \in \mathbb{R}^m \\ x^u \in \mathbb{R}^{n-m}}}{\text{minimize}} \quad f(x^y, x^u) \quad (15.4.175a)$$

$$\text{subject to} \quad c(x^y, x^u) = 0 \quad (15.4.175b)$$

$$\text{and} \quad a \leq x^u \leq b, \quad (15.4.175c)$$

where the problem variables have been partitioned as  $x = (x^y, x^u)$ ; for optimal control problems,  $x^y$  are the state variables,  $x^u$  are the control variables, and (15.4.175b) are the state equations. Heinkenschloss and Vicente (1995), Vicente (1995), Dennis, Heinkenschloss, and Vicente (1998), and Heinkenschloss and Vicente (1999) developed methods that are hybrids of the Coleman–Li approach of Section 13.12 to deal with the simple bounds, and the Byrd–Omojokun approach of Section 15.4.2 to handle the equality constraints. The tangential-step subproblem (15.4.22) exploits the fact that the null-space basis matrix

$$N(x) = \begin{pmatrix} -\nabla_{x^y} c(x)^{-1} \nabla_{x^u} c(x) \\ I_{n-m} \end{pmatrix}$$

(see (4.4.13) [p. 72]) is particularly easy to manipulate for optimal-control-like applications.

To simplify matters further, the tangential-step trust-region bound

$$\|N(x)t^N\| = \left\| \begin{pmatrix} -\nabla_{x^y} c(x)^{-1} \nabla_{x^u} c(x)t^N \\ t^N \end{pmatrix} \right\| \leq \xi^T \Delta$$

is often replaced by the less restrictive  $\|t^N\| \leq \xi^T \Delta$ . Finally, approximations to the matrices involved are allowed and include the possibility that the tangential step may move away (slightly) from the null-space of  $\nabla_x c(x)$ . The resulting algorithm can be shown to converge globally and Q-superlinearly to a second-order critical point under standard assumptions; a software package known as TRICE provides an implementation of this algorithm. This framework has been extended in the obvious way by Das (1996) to cover the case where there are bounds on all variables. Significantly, he observes in practice that such methods appear to be extremely sensitive to the values of the trust-region relaxation parameter  $\xi^N$  and to the initial trust-region radius.

The method proposed by Yamashita, Yabe, and Tanabe (1997) is much closer in spirit to Vardi's method. For simplicity, the constraints may be considered to be equalities  $c(x) = 0$  and simple bounds (in this case, nonnegativities)  $x \geq 0$ ; the simple bounds are treated via logarithmic barrier terms, while the other constraints are penalized using  $\ell_1$  penalty terms, leading to the penalty barrier function

$$f(x) + \sigma \|c(x)\|_1 - \mu \langle e, \log(x) \rangle.$$

Firstly, a trial Newton primal-dual step of the form

$$\begin{pmatrix} H & A^T(x) & -I \\ A(x) & 0 & 0 \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} s^T \\ -s^y \\ s^z \end{pmatrix} = - \begin{pmatrix} \nabla_x \ell(x, y) - z \\ c(x_k) \\ \mu - Xz \end{pmatrix} \quad (15.4.176)$$

(which is a combination of (13.6.2) [p. 519] for the nonnegativities and (15.2.3) for the equations) is found, where  $H$  is a (fixed) positive definite diagonal matrix. Here  $y$  and  $z$  are estimates of the Lagrange multipliers for the equality constraints and the dual variables associated with the simple bounds, respectively. Having computed  $s^T$ , a Cauchy step  $s^C = \alpha s^T$  is then found by retreating (if necessary) along  $s^T$  to minimize the quadratic model

$$m(s) = f(x) - \mu\langle e, \log(x) \rangle + \langle \nabla_x \ell(x, y) - \mu X^{-1}e, s \rangle + \frac{1}{2}\langle s, Hs \rangle \sigma \|c(x) + A(x)s\|$$

of the combined penalty barrier function within  $\mathcal{B}(x, s, \Delta)$ , which is the intersection of the “sufficiently feasible” region for the simple bounds,  $\{s \mid x + s \geq \varsigma_k x\}$ , and the trust region. The actual step used is then any step that lies within  $\mathcal{B}(x, s, \Delta)$  giving at least a fixed fraction of the decrease in  $m(s)$  achieved for the Cauchy step. Ideally, the full Newton primal-dual step, that is, (15.4.176) with  $H$  being a good approximation to the Hessian of the Lagrangian, will prove suitable. The step is accepted or rejected, and the radius updated, in the time-honoured trust-region way by comparing the model reduction with that actually achieved for the penalty-barrier function. To avoid the Maratos effect, a nonmonotone strategy is used, but as we have already seen this has essentially the same effect as computing a second-order correction. Yamashita, Yabe, and Tanabe (1997) show that this is a most effective method in practice and build on the convergence results of Yamashita and Yabe (1996a, 1996b) and Yabe and Yamashita (1997) to show that their algorithm is both globally and Q-superlinearly convergent under standard assumptions.

See also Fan, Sarkar, and Lasdon (1988), Ternet (1994), and Chen and Han (1996).

## 15.5 The Filter Method

We now turn our attention to a very recent SQP method for solving the general nonlinear programming problem (15.1.1). The aim of the technique described in what follows is to dispense with the need for a penalty function, a feature common to all the algorithms we have described so far in the chapter. The advantage of doing so is that this saves us from having to choose an appropriate initial value for the penalty parameter, and from designing a procedure to update the parameter if this proves necessary. Instead, the idea will be to use a “filter”, which we describe below, hence the name “filter method”. This filter will interfere as little as possible with the Newton iteration implied by SQP, which we hope will encourage fast convergence.

### 15.5.1 A Composite-Step Approximate SQP Framework

According to the distinctions made at the beginning of this chapter, the filter method may be classified as an IQP method, because, at iteration  $k$ , it attempts to solve the subproblem

$$\begin{aligned} & \text{minimize} && m_k(s) \\ & \text{subject to} && c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s = 0, \\ & && c_{\mathcal{T}}(x_k) + A_{\mathcal{T}}(x_k)s \geq 0, \end{aligned} \quad \left. \right\} \text{QP}(x_k)$$

where

$$m_k(s) = f(x_k) + \langle g_k, s \rangle + \frac{1}{2}\langle s, H_k s \rangle,$$

with  $H_k$  being a symmetric approximation of the Hessian of the Lagrangian function  $\ell(x, y) = f(x) + \langle y, c(x) \rangle$  at  $x_k$  for some value of the Lagrange multipliers  $y_k$ . More precisely, we let

$$H_k = B_k + \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_k]_i W_{ik},$$

where

$$B_k \approx \nabla_{xx} f(x_k) \text{ and } W_{ik} \approx \nabla_{xx} c_i(x_k).$$

In the by-now-familiar manner, we add the trust-region constraint to the subproblem, which then becomes

$$\begin{aligned} & \text{minimize} && m_k(s) \\ & \text{subject to} && \left. \begin{array}{l} c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s = 0, \\ c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s \geq 0, \\ \|s\| \leq \Delta_k. \end{array} \right\} \text{TRQP}(x_k, \Delta_k) \end{aligned}$$

Suppose, for now, that this subproblem has a nonempty feasible set and a solution  $s_k$ . Once again, we follow the composite-step paradigm by noting that the step  $s_k$  may be viewed as the sum of a normal step  $n_k$ , such that  $n_k$  satisfies the constraints of  $\text{TRQP}(x_k, \Delta_k)$ , and a tangential step  $t_k$ , whose purpose is to obtain reduction of the model while continuing to satisfy those constraints. More formally, we write

$$s_k = n_k + t_k$$

and assume that

$$c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)n_k = 0, \quad c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)n_k \geq 0, \quad (15.5.1a)$$

$$\|s_k\| \leq \Delta_k, \quad \text{and} \quad (15.5.1b)$$

$$c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s_k = 0, \quad c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s_k \geq 0. \quad (15.5.1c)$$

There are, of course, various ways to compute  $n_k$  and  $t_k$ . For instance, we could compute  $n_k$  as

$$n_k = P_k[x_k] - x_k, \quad (15.5.2)$$

where  $P_k$  is the orthogonal projector onto the feasible set of  $\text{QP}(x_k)$ . In what follows, we do not make any specific choice for  $n_k$ , but we shall make the assumptions that  $n_k$  exists when the maximum violation of the nonlinear constraints at the  $k$ th iterate  $\vartheta_k \stackrel{\text{def}}{=} \vartheta(x_k)$  is sufficiently small, where

$$\vartheta(x) \stackrel{\text{def}}{=} \|c(x)^{\mathcal{I}^-}\| \equiv \max \left[ 0, \max_{i \in \mathcal{E}} |c_i(x)|, \max_{i \in \mathcal{I}} -c_i(x) \right], \quad (15.5.3)$$

and that  $n_k$  is reasonably scaled with respect to the values of the constraints. More formally, we assume the following.

**AA.11** There exists a constant<sup>260</sup>  $\kappa_{\text{usc}} > 0$  such that, if  $\{x_{k_i}\}$  is any subsequence of iterates with

$$\lim_{i \rightarrow \infty} \vartheta_{k_i} = 0,$$

---

<sup>260</sup> “usc” stands for “upper bound on the step with respect to the constraint”.

then  $n_{k_i}$  exists for  $i$  sufficiently large and

$$\|n_{k_i}\| \leq \kappa_{\text{usc}} \vartheta_{k_i}. \quad (15.5.4)$$

Note the similarity between this assumption and AA.1g, but note also that AA.1l only requires (15.5.4) to hold when the constraint violation is sufficiently small, which is a far weaker requirement. We can also view AA.1l in terms of the constraint functions themselves and the geometry of the boundary of the feasible set. For instance, if we define

$$\mathcal{F}(x) \stackrel{\text{def}}{=} \{v \in \mathbb{R}^n \mid c_{\mathcal{E}}(x) + A_{\mathcal{E}}(x)(v - x) = 0, \quad c_{\mathcal{I}}(x) + A_{\mathcal{I}}(x)(v - x) \geq 0\}$$

and assume that, at every limit point  $x_*$  of the sequence of iterates, the relative interior of the linearized constraints  $\text{ri}\{\mathcal{F}(x_*)\}$  is nonempty, we know, by applying a continuity argument, that the same must be true for any iterate  $x_k$  that is close enough to  $x_*$ . Hence the feasible set of  $\text{QP}(x_k)$  is nonempty for such an  $x_k$ , which implies that  $P_k$  is well defined and that a normal step  $n_k$  of the form (15.5.2) exists. Furthermore, if AO.1b holds, that is, if the singular values of the Jacobian of constraints active at  $x_*$ ,  $A_{\mathcal{A}(x_*)}(x_*)$ , are bounded away from zero, the same must be true by continuity for  $A_{\mathcal{A}(x_*)}(x_k)$ , and the projection operator

$$P_k = A_{\mathcal{A}(x_*)}^T(x_k) \left[ A_{\mathcal{A}(x_*)}(x_k) A_{\mathcal{A}(x_*)}^T(x_k) \right]^{-1} A_{\mathcal{A}(x_*)}(x_k)$$

must be bounded in norm for all  $x_k$  sufficiently close to  $x_*$ . Since only the constraints active at  $x_*$  can be active in a sufficiently small neighbourhood of this limit point, the boundedness of the projection operator in turn guarantees that (15.5.4) holds for the normal step

$$-A_{\mathcal{A}(x_*)}^T(x_k) \left[ A_{\mathcal{A}(x_*)}(x_k) A_{\mathcal{A}(x_*)}^T(x_k) \right]^{-1} c_{\mathcal{A}(x_*)}(x_k)$$

for all  $k$  sufficiently large, because AI.1 ensures that  $x_k$  must be arbitrarily close to at least one limit point of the sequence of iterates for such  $k$ . Note that this assumption is considerably weaker than AA.14, for instance. Thus we see that AA.1l does not impose conditions on the constraints or the normal step itself that are unduly restrictive.

Having defined the normal step  $n_k$ , we write

$$x_k^N = x_k + n_k$$

and observe that  $n_k$  satisfies the constraints of  $\text{QP}(x_k)$ . It is crucial to note, at this stage, that  $n_k$  may fail to exist because the constraints of  $\text{QP}(x_k)$  may be incompatible, in which case  $P_k$  is undefined.

Let us continue to consider the case where this failure does not arise and a normal step  $n_k$  has been found. If  $x_k^N$  lies in the trust region

$$\mathcal{B}_k = \{x \mid \|x - x_k\| \leq \Delta_k\},$$

which implies that  $n_k$  also satisfies the constraints of  $\text{TRQP}(x_k, \Delta_k)$ , we then have to find a tangential step  $t_k$ , starting from  $x_k^N$  and satisfying (15.5.1), with the aim of

decreasing the value of the objective function. As always in trust-region methods, this is achieved by computing a step that produces a sufficient decrease in  $m_k$ , which is to say that we wish  $m_k(x_k^N) - m_k(x_k + s_k)$  to be “sufficiently” large. Of course, this is only possible if the maximum size of  $t_k$  permitted by the trust-region radius  $\Delta_k$  is not too small, which is to say that  $x_k^N$  is not too close to the trust-region boundary. We formalize this condition by requiring that

$$\|n_k\| \leq \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^\mu] \quad (15.5.5)$$

for some  $\kappa_\Delta \in (0, 1)$ , some  $\kappa_\mu > 0$ , and some  $\mu \in (0, 1)$ . If condition (15.5.5) does not hold, we assume that the computation of  $t_k$  is unlikely to produce a satisfactory decrease in  $m_k$  and proceed just as if the feasible set of  $\text{TRQP}(x_k, \Delta_k)$  were empty. If  $n_k$  can be computed and (15.5.5) holds, we shall say that  $\text{TRQP}(x_k, \Delta_k)$  is *compatible*; in this case, at least, a sufficient decrease seems possible. In order to formalize what we mean, we recall that the feasible set of  $\text{TRQP}(x_k, \Delta_k)$  is convex, and we can therefore use the first-order criticality measure

$$\chi_k = \chi(x_k) = \left| \min_{\substack{c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)d=0 \\ c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)d \geq 0 \\ \|d\| \leq 1}} \langle g(x_k^N), d \rangle \right|,$$

which we introduced in Theorem 12.1.6 (p. 449). Note that this function is continuous in its argument, as we required in (8.1.2) (p. 250) for it to be a criticality measure, because both the gradient of the objective function and the Jacobian of the constraints are continuous (assuming AW.1). We also observe that  $\chi_k = 0$  when  $x_k^N$  is a first-order critical point of  $\text{TRQP}(x_k, \Delta_k)$  (see Chapter 12), and if

$$x_k = x_k^N \text{ and } \chi_k = 0,$$

$x_k$  is itself a first-order critical point for the original problem (15.1.1).

Having already considered the conditions we require for the normal step, we now formulate our requirement that the tangential step yield a sufficient model decrease in the form of a by-now-familiar Cauchy-point condition (where  $\beta_k = 1 + \|H_k\|$ ).

**AA.1m** There exists a constant<sup>261</sup>  $\kappa_{\text{tmd}} > 0$  such that

$$m_k(x_k^N) - m_k(x_k^N + t_k) \geq \kappa_{\text{tmd}} \chi_k \min \left[ \frac{\chi_k}{\beta_k}, \Delta_k \right].$$

Again, note the similarity in spirit of AA.1m and AA.1i.

As we saw in Theorem 12.2.2 (p. 457), this condition holds if the model reduction exceeds that which would be obtained at the generalized Cauchy point, which is the point resulting from the application of Algorithm 12.2.2 along the projected gradient path from  $x_k^N$ , that is,

$$x_k(\alpha) = P_k[x_k^N - \alpha \nabla_x m_k(x_k^N)].$$

---

<sup>261</sup>“tmd” stands for “tangential minimum decrease”.

Let us now return to the case where  $\text{TRQP}(x_k, \Delta_k)$  is not compatible, that is, when the feasible set determined by the constraints of  $\text{QP}(x_k)$  is empty or the freedom left to reduce  $m_k$  within the trust region is too small in the sense that (15.5.5) fails. In this situation, solving  $\text{TRQP}(x_k, \Delta_k)$  is most likely pointless, and we must consider an alternative. We base this on the intuitive observation that, if  $c(x_k)$  is sufficiently small and the true nonlinear constraints are locally compatible, the linearized constraints should also be compatible, since they approximate the nonlinear constraints (locally) correctly. Furthermore, the feasible region for the linearized constraints should then be close enough to  $x_k$  for there to be some room to reduce  $m_k$ , at least if  $\mathcal{B}_k$  is large enough. If the nonlinear constraints are locally incompatible, we have to find a neighbourhood where this is not the case, since the problem (15.1.1) does not otherwise make sense. We thus rely on a *restoration procedure*, whose aim is to produce a new point  $x_k + r_k$  for which  $\text{TRQP}(x_k + r_k, \Delta_{k+1})$  is compatible for some  $\Delta_{k+1} > 0$ . We will actually require an additional condition, which we will discuss shortly.

The idea of the restoration procedure is to (approximately) solve

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \vartheta(x) \tag{15.5.6}$$

starting from  $x_k$ , the current iterate. This is a nonsmooth problem of the type we have already studied in Chapter 11 and will consider again in Section 16.2, and we have seen that Algorithm 11.1.1 (p. 413) can be successfully applied to solve it. Thus we will not describe the restoration procedure in detail. Note that we have chosen here to reduce the infinity norm of the constraint violation, but we could equally well consider other norms, such as  $\ell_1$  or  $\ell_2$ . Of course, this technique only guarantees convergence to a first-order critical point of the chosen measure of constraint violation, which means that, in fact, the restoration procedure may fail, as this critical point may not be feasible for the constraints of (15.1.1). However, even in this case, the result of the procedure is of interest because it typically produces a local minimizer of  $\vartheta(x)$ , or of whatever other measure of constraint violation we choose for the restoration, yielding a point of locally-least infeasibility.

There is no easy way to circumvent this drawback, as it is known that finding a feasible point can be just as difficult as the optimization problem (15.1.1) itself.<sup>262</sup> We therefore accept two possible outcomes of the restoration procedure: either the procedure fails in that it does not produce a sequence of iterates converging to feasibility, or a point  $x_k + r_k$  is produced such that  $\vartheta(x_k + r_k)$  is as small as we wish. We will shortly see that this is all we need.

### 15.5.2 The Notion of a Filter

Having computed a step  $s_k = n_k + t_k$  (or  $r_k$ ), we still need to decide whether the trial point  $x_k + s_k$  (or  $x_k + r_k$ ) is any better than  $x_k$  as an approximate solution to our

---

<sup>262</sup>In practice, this is rarely the case, since the solution set for the former is almost always far bigger than that for the latter.

original problem (15.1.1). As we have discussed before, one way of handling the possibly conflicting goals of feasibility and objective function decrease is to combine them to produce a single objective, using a penalty function. Although we have described a number of ways of using such functions, there is always an unspoken assumption that finding and manipulating the associated penalty parameter is easy. Since this is not always the case, we shall pursue a different approach here.

We shall use a concept borrowed from multicriteria optimization. We say that a point  $x_1$  *dominates* a point  $x_2$  whenever

$$\vartheta(x_1) \leq \vartheta(x_2) \text{ and } f(x_1) \leq f(x_2).$$

Thus, if iterate  $x_k$  dominates iterate  $x_j$ , the latter is of no real interest to us since  $x_k$  is at least as good as  $x_j$  on account of both feasibility and optimality. All we need to do now is to remember iterates that are not dominated by any other iterates using a structure called a filter. A *filter* is a list  $\mathcal{F}$  of pairs of the form  $(\vartheta_i, f_i)$  such that

$$\vartheta_i < \vartheta_j \text{ or } f_i < f_j$$

for  $i \neq j$ . We thus aim to accept a new iterate  $x_i$  only if it is not dominated by any other iterate in the filter. In the vocabulary of multicriteria optimization, this amounts to building elements of the efficient frontier associated with the bicriteria problem of reducing infeasibility and the objective function value. Figure 15.5.1 illustrates the concept of a filter by showing the pairs  $(\vartheta_k, f_k)$  as black dots in the  $(\vartheta, f)$  space. Each such pair is called the  $(\vartheta, f)$ -pair associated with  $x_k$ . The lines radiating from each  $(\vartheta, f)$ -pair indicate that any iterate whose associated  $(\vartheta, f)$ -pair occurs above and to the right of that of a given filter point is dominated by this  $(\vartheta, f)$ -pair.

While the idea of not accepting dominated trial points is simple and elegant, it needs to be refined a little in order to provide an efficient algorithmic tool. In particular, we do not wish to accept  $x_k + s_k$  if its  $(\vartheta, f)$ -pair is arbitrarily close to that of  $x_k$  or that of a point already in the filter. Thus we set a small “margin” around the border of the dominated part of the  $(\vartheta, f)$ -space in which we shall also reject trial points. Formally, we say that a point  $x$  is *acceptable for the filter* if and only if

$$\vartheta(x) < (1 - \gamma_\vartheta)\vartheta_j \text{ or } f(x) < f_j - \gamma_\vartheta\vartheta(x) \text{ for all } (\vartheta_j, f_j) \in \mathcal{F} \quad (15.5.7)$$

for some  $\gamma_\vartheta \in (0, 1)$ . In Figure 15.5.1, the set of acceptable points corresponds to the set of  $(\vartheta, f)$ -pairs below the filter envelope defined by the thinner line (the envelope has been enlarged on the figure for clarity). We also say that  $x$  is “acceptable for the filter and  $x_k$ ” if (15.5.7) holds with  $\mathcal{F}$  replaced by  $\mathcal{F} \cup (\vartheta_k, f_k)$ . We thus move from  $x_k$  to  $x_k + s_k$  only if  $x_k + s_k$  is acceptable for the filter and  $x_k$ .

As the algorithm progresses, we may want to *add a  $(\vartheta, f)$ -pair to the filter*. If an iterate  $x_k$  is acceptable for  $\mathcal{F}$ , we do this by adding the pair  $(\vartheta_k, f_k)$  to the filter and by removing from it every other pair  $(\vartheta_j, f_j)$  such that  $\vartheta_j \geq (1 - \gamma_\vartheta)\vartheta_k$  and  $f_j \geq f_k - \gamma_\vartheta\vartheta_j$ . We also refer to this operation as “adding  $x_k$  to the filter” although, strictly speaking, it is the  $(\vartheta, f)$ -pair that is added.

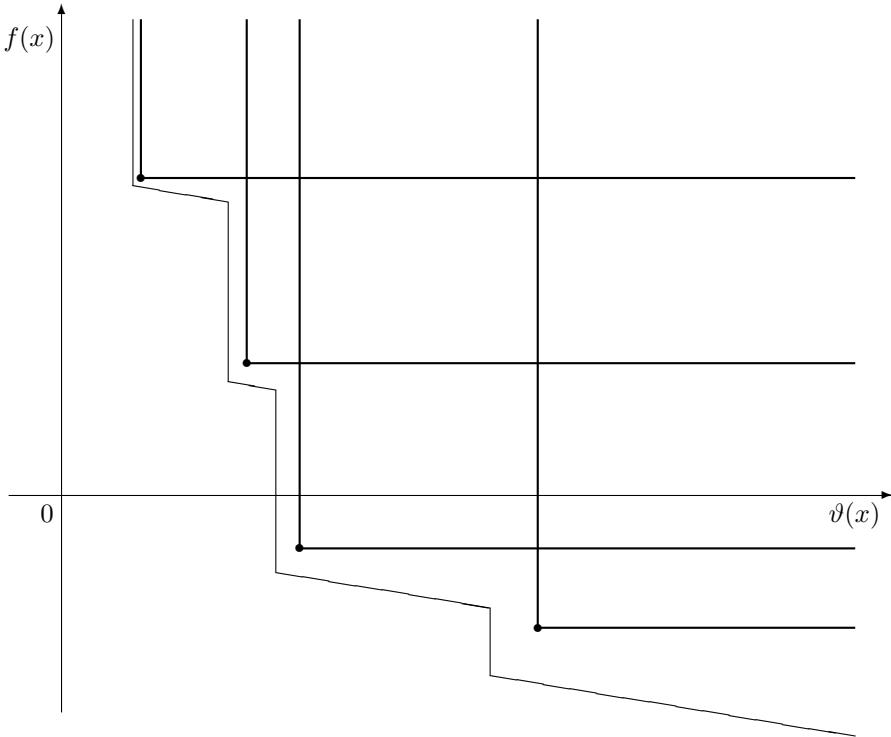


Figure 15.5.1: A filter with four pairs.

### 15.5.3 An SQP Filter Algorithm

We have discussed the main ingredients of the class of algorithms we wish to consider and we are now ready to define it formally.

#### Algorithm 15.5.1: SQP filter algorithm

**Step 0: Initialization.** Let an initial point  $x_0$ , an initial trust-region radius  $\Delta_0 > 0$ , an initial vector of Lagrange multipliers  $y_0$ , and initial symmetric approximations  $B_0$  and  $\{W_{i,0}\}_{i \in \mathcal{E} \cup \mathcal{I}}$  be given, as well as constants  $\gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$ ,  $0 < \eta_1 \leq \eta_2 < 1$ ,  $\gamma_\vartheta \in (0, 1)$ ,  $\kappa_\vartheta \in (0, 1)$ ,  $\kappa_\Delta \in (0, 1]$ ,  $\kappa_\mu > 0$ ,  $\mu \in (0, 1)$ , and  $\kappa_{\text{tmd}} \in (0, 1]$ . Compute  $f(x_0)$  and  $c(x_0)$  and set  $\mathcal{F} = \emptyset$  and  $k = 0$ .

**Step 1: Ensure compatibility.** Attempt to compute a step  $n_k$ . If TRQP  $(x_k, \Delta_k)$  is compatible, go to Step 2. Otherwise, include  $x_k$  in the filter and compute a restoration step  $r_k$  for which TRQP  $(x_k + r_k, \Delta_{k+1})$  is compatible for some  $\Delta_{k+1} > 0$ , and  $x_k + r_k$  is acceptable for the filter. If this proves impossible, stop. Otherwise, define  $x_{k+1}$  to be  $x_k + r_k$  and go to Step 6.

**Step 2: Determine a trial step.** Compute a step  $t_k$  for which AA.1m holds and set  $s_k = n_k + t_k$ .

**Step 3: Tests to accept the trial step.**

- Evaluate  $c(x_k + s_k)$  and  $f(x_k + s_k)$ .
- If  $x_k + s_k$  is not acceptable for the filter and  $x_k$ , set  $x_{k+1} = x_k$ , choose  $\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$ , increment  $k$  by 1, and go to Step 1.
- If

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\vartheta \vartheta_k^2 \quad (15.5.8)$$

and

$$\rho_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} < \eta_1, \quad (15.5.9)$$

again set  $x_{k+1} = x_k$ , choose  $\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$ , increment  $k$  by 1, and go to Step 1.

**Step 4: Test to include the current iterate in the filter.** If (15.5.8) fails, include  $x_k$  in the filter  $\mathcal{F}$ .

**Step 5: Move to the new iterate.** Set  $x_{k+1} = x_k + s_k$  and choose

$$\Delta_{k+1} \in \begin{cases} [\gamma_1 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k \geq \eta_2. \end{cases}$$

**Step 6: Update the approximations.** Determine  $B_{k+1}$ ,  $\{W_{i,k+1}\}_{i \in \mathcal{E} \cup \mathcal{I}}$ , and  $y_{k+1}$ . Increment  $k$  by 1 and go to Step 1.

Reasonable values for the constants might be

$$\begin{aligned} \gamma_0 &= 0.1, & \gamma_1 &= 0.5, & \gamma_2 &= 2, & \eta_1 &= 0.01, & \eta_2 &= 0.9, \\ \gamma_\vartheta &= 10^{-4}, & \kappa_\Delta &= 0.7, & \kappa_\mu &= 100, & \mu &= 0.01, & \kappa_\vartheta &= 10^{-4}, \text{ and } \kappa_{\text{tmd}} = 0.01, \end{aligned}$$

but it is too early to know if these are even close to the best possible choices.

Observe first that, by construction, every iterate  $x_k$  must be acceptable for the filter at the beginning of iteration  $k$ , irrespective of the possibility that it is added to the filter later. Also note that the mechanism of the algorithm prevents two successive restoration steps, since such a step  $r_k$  makes  $\text{TRQP}(x_k + r_k, \Delta_{k+1})$  compatible. Moreover,  $r_k$  cannot be zero, that is, restoration cannot simply entail enlarging the trust-region radius to ensure (15.5.5), even if  $n_k$  exists. This is because  $x_k$  is added to the filter before  $r_k$  is computed, and  $x_k + r_k$  must be acceptable for the filter that now contains  $x_k$ .

For the restoration procedure in Step 1 to succeed, we have to evaluate whether  $\text{TRQP}(x_k + r_k, \Delta_{k+1})$  is compatible for a suitable value of  $\Delta_{k+1}$ . This requires that a suitable normal step be computed that successfully passes the test (15.5.5). Of course,

once this is achieved, this normal step may be reused at iteration  $k + 1$ . Thus we shall require that the normal step calculated to verify compatibility of  $\text{TRQP}(x_k + r_k, \Delta_{k+1})$  should actually be used as  $n_{k+1}$ .

As it stands, the algorithm is not specific about how to choose  $\Delta_{k+1}$  during a restoration iteration. On the one hand, there is an advantage to choosing a large  $\Delta_{k+1}$ , since this allows a large step and, one hopes, good progress. On the other hand, it may be unwise to choose it to be too large, as this may result in a large number of unsuccessful iterations, during which the radius is reduced, before the algorithm can make any progress. A reasonable choice might be to restart from the average radius observed during past successful iterations. Or one could apply the internal doubling strategy of Section 10.5.1 to increase the new radius, or even consider the technique described in Section 17.2. However, we recognize that numerical experience with the algorithm is too limited at this stage to make definite recommendations.

If the iterate  $x_k$  is feasible, then  $x_k = x_k^N$  and we obtain that

$$0 = \kappa_\vartheta \vartheta_k^2 \leq m_k(x_k^N) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k + s_k).$$

As a consequence *no feasible iterate is ever included in the filter*, which is crucial in allowing finite termination of the restoration procedure. Indeed, if the restoration procedure is required at iteration  $k$  of the filter algorithm and produces a sequence of points  $\{x_{k,j}\}$  converging to feasibility, there must be an iterate  $x_{k,j}$  for which

$$\vartheta(x_{k,j}) \leq \min \left[ (1 - \gamma_\vartheta) \vartheta_k^{\min}, \frac{\kappa_{\text{usc}}}{\kappa_\Delta} \Delta_{k+1} \min[1, \kappa_\mu \Delta_{k+1}^\mu] \right]$$

for any given  $\Delta_{k+1} > 0$ , where

$$\vartheta_k^{\min} = \min_{i \in \mathcal{A}, i \leq k} \vartheta_i > 0$$

and

$$\mathcal{A} = \{k \mid x_k \text{ is added to the filter}\}.$$

Moreover,  $\vartheta_{k,j}$  must eventually be small enough to ensure the existence of a normal step from  $x_{k,j}$ , provided that AA.11 also holds for the restoration iterates  $x_{k,j}$ . In other words, the restoration iteration must eventually find an iterate  $x_{k,j}$  which is acceptable for the filter and for which the normal step exists and, because of AA.11, satisfies (15.5.5), that is, an iterate  $x_j$  that is both acceptable and compatible. As a consequence, the restoration procedure will terminate in a finite number of steps, and the filter algorithm may then proceed.

Notice that (15.5.8) ensures that the denominator of  $\rho_k$  in (15.5.9) will be strictly positive whenever  $\vartheta_k$  is. If  $\vartheta_k = 0$ , then  $x_k = x_k^N$ , and the denominator of (15.5.9) will be strictly positive unless  $x_k$  is a first-order critical point because of AA.1m.

Finally, we recognize that AA.1m may be difficult to verify in practice, since it may be expensive to compute  $x_k^N$  and  $P_k$ . We shall consider a possibly cheaper alternative in Section 15.5.5.

### 15.5.4 Convergence to First-Order Critical Points

We now prove that our algorithm generates a globally convergent sequence of iterates, at least if the restoration iteration always succeeds. For the purpose of our analysis, we shall consider

$$\mathcal{S} = \{k \mid x_{k+1} = x_k + s_k\}$$

the set of (indices of) successful iterations, and

$$\mathcal{R} = \{k \mid n_k \text{ does not exist or } \|n_k\| > \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^\mu]\}$$

the set of *restoration* iterations; we shall refer to those iterations whose indices do not lie in  $\mathcal{R}$  as *normal* iterations. In order to obtain our global convergence result, we will use the assumptions AW.1 (smoothness of the objective function and constraints), AA.1l and AA.1m, AM.4j (boundedness of the matrices  $H_k$ ), and AI.1 (boundedness of the sequence of iterates). We specify the bound implied by AM.4j as

$$\|H_k\| \leq \kappa_{\text{umh}} - 1 < \kappa_{\text{umh}} \text{ for all } k,$$

for some  $\kappa_{\text{umh}} > 1$ , and observe that this inequality holds if, for example, the Lagrange multipliers  $y_k$  and the Hessian approximations  $\{W_{ik}\}_{i \in \mathcal{E} \cup \mathcal{T}}$  and  $B_k$  remain bounded. A first immediate consequence of AW.1 is that there exists a constant  $\kappa_{\text{ubh}} > 1$  such that, for all  $k$ ,

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\text{ubh}} \Delta_k^2, \quad (15.5.10)$$

as is implied by Theorem 6.4.1 (p. 133). A second important consequence of our assumptions is that AW.1 and AI.1 together directly ensure that, for all  $k$ ,

$$f^{\min} \leq f(x_k) \leq f^{\max} \text{ and } 0 \leq \vartheta_k \leq \vartheta^{\max} \quad (15.5.11)$$

for some constants  $f^{\min}$ ,  $f^{\max}$ , and  $\vartheta^{\max} > 0$ . Thus the part of the  $(\vartheta, f)$ -space in which the  $(\vartheta, f)$ -pairs associated with the filter iterates lie is restricted to the rectangle

$$\mathcal{A}_0 = [0, \vartheta^{\max}] \times [f^{\min}, f^{\max}],$$

whose area,  $\text{surf}(\mathcal{A}_0)$ , is clearly finite. If there are  $(\vartheta, f)$ -pairs in the filter  $\mathcal{F}$  at iteration  $k$ , we let  $\mathcal{A}_k$  be the part of  $\mathcal{A}_0$  in which the  $(\vartheta, f)$ -pairs associated with a new iterate must fall for this iterate to be acceptable, that is,

$$\mathcal{A}_k = \{(\vartheta(x), f(x)) \in \mathcal{A}_0 \mid (15.5.7) \text{ holds}\}.$$

We also note the following simple consequence of AA.1l and AI.1.

**Lemma 15.5.1** Suppose that AA.1l and AI.1 hold and that  $\{x_{k_i}\}$  is a subsequence of iterates for which

$$\lim_{i \rightarrow \infty} \vartheta_{k_i} = 0.$$

Then there exists a constant<sup>263</sup>  $\kappa_{\text{lsc}} > 0$  such that

$$\kappa_{\text{lsc}} \vartheta_{k_i} \leq \|n_{k_i}\| \quad (15.5.12)$$

for  $i$  sufficiently large.

**Proof.** Consider an iterate  $x_{k_i}$  for which  $\vartheta_{k_i} > 0$  and for which  $n_{k_i}$  exists (as a consequence of AA.1l), and define

$$\mathcal{V}_{k_i} \stackrel{\text{def}}{=} \{j \in \mathcal{E} \mid \vartheta_{k_i} = |c_j(x_{k_i})|\} \bigcup \{j \in \mathcal{I} \mid \vartheta_{k_i} = -c_j(x_{k_i})\},$$

which is the subset of the most-violated constraints. From the definitions of  $\vartheta_{k_i}$  in (15.5.3) and of the normal step in (15.5.1a) we obtain, using the Cauchy–Schwartz inequality, that

$$\vartheta_{k_i} \leq |\langle \nabla_x c_j(x_{k_i}), n_{k_i} \rangle| \leq \|\nabla_x c_j(x_{k_i})\| \|n_{k_i}\| \quad (15.5.13)$$

for all  $j \in \mathcal{V}_{k_i}$ . But AI.1 ensures that there exists a constant  $\kappa_{\text{lsc}} > 0$  such that

$$\max_{j \in \mathcal{E} \cup \mathcal{I}} \max_{x \in \mathcal{X}} \|\nabla_x c_j(x)\| \stackrel{\text{def}}{=} \frac{1}{\kappa_{\text{lsc}}},$$

where  $\mathcal{X}$  is a compact subset of  $\mathbb{R}^n$  containing all the iterates. We then obtain the desired conclusion by substituting this bound in (15.5.13).  $\square$

We start our analysis by examining what happens when an infinite number of iterates (that is, their  $(\vartheta, f)$ -pairs) are added to the filter.

**Lemma 15.5.2** Suppose that AW.1 and AI.1 hold and that  $\{k_i\}$  is any infinite subsequence at which the iterate  $x_{k_i}$  is added to the filter. Then

$$\lim_{i \rightarrow \infty} \vartheta_{k_i} = 0.$$

**Proof.** Suppose, for the purpose of obtaining a contradiction, that there exists an infinite subsequence  $\{k_j\} \subseteq \{k_i\}$  for which

$$\vartheta_{k_j} \geq \epsilon \quad (15.5.14)$$

---

<sup>263</sup> “lsc” stands for “lower bound on the step with respect to the constraint”.

for some  $\epsilon > 0$ . At each iteration  $k_j$ , the  $(\vartheta, f)$ -pair associated with  $x_{k_j}$ , that is,  $(\vartheta_{k_j}, f_{k_j})$ , is added to the filter. This means that, as long as the pair  $(\vartheta_{k_j}, f_{k_j})$  remains in the filter, no other  $(\vartheta, f)$ -pair can be added to the filter within the triangle

$$\{(\vartheta, f) \mid \vartheta \leq \vartheta_{k_j}, f \leq f_{k_j} \text{ and } f \geq f_{k,j} - \vartheta + (1 - \gamma_\vartheta)\vartheta_{k_j}\}, \quad (15.5.15)$$

or within the intersection of this triangle with  $\mathcal{A}_0$ . Assume now that this pair is dominated by another pair  $(\vartheta_{k_j+\ell}, f_{k_j+\ell})$ , which is included in the filter at some subsequent iteration  $k_j + \ell$  ( $\ell > 0$ ). This means that  $f_{k_j+\ell} \leq f_{k_j}$ , which in turn ensures that the triangle (15.5.15) is entirely above the line  $f_{k_j+\ell} - \gamma_\vartheta\vartheta$ . Furthermore, we also have that  $\vartheta_{k_j+\ell} \leq \vartheta_{k_j}$ , which then implies that the triangle (15.5.15) lies entirely in the part of the  $(\vartheta, f)$  plane where iterates are not accepted (this situation is illustrated in Figure 15.5.2 for the worst possible case where  $f_{k_j+\ell} = f_{k_j}$ ). Thus no other  $(\vartheta, f)$ -pair can ever be added to the filter within this triangle. But the area of the triangle is at least  $\frac{1}{2}\gamma_\vartheta^2\epsilon^2$ . Thus the set  $\mathcal{A}_0$  is completely covered by at most  $\lceil 2\text{surf}(\mathcal{A}_0)/\gamma_\vartheta^2\epsilon^2 \rceil$  such triangles, which puts a finite upper bound on the number of iterations in  $\{k_j\}$ . Hence (15.5.14) is impossible for any infinite subsequence of  $\{k_i\}$ , and the conclusion follows.  $\square$

We next examine the size of the constraint violation before and after a normal iteration.

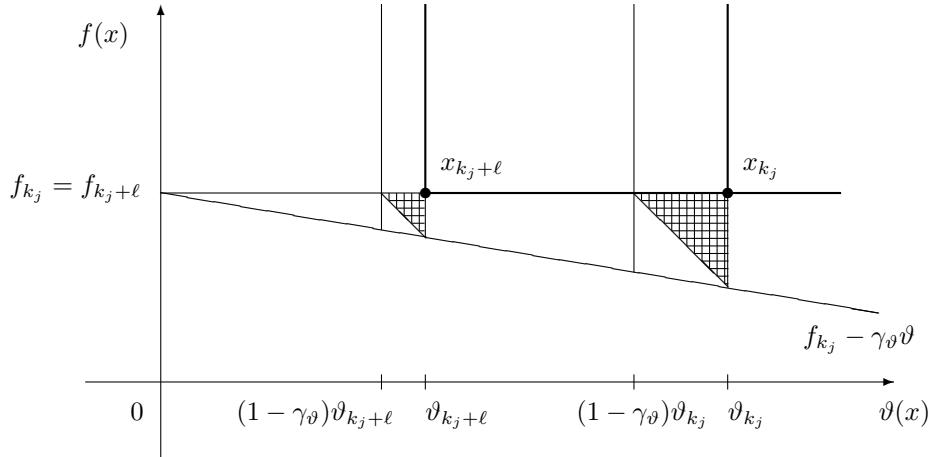


Figure 15.5.2: The appearance of the filter when iterate  $x_{k_j+\ell}$  dominates  $x_{k_j}$  and the triangles of the proof of Lemma 15.5.2. Note that both triangles remain above the line  $f = f_{k_j} - \gamma_\vartheta\vartheta$ .

**Lemma 15.5.3** Suppose that AW.1 and AI.1 hold, that  $k \notin \mathcal{R}$ , and that  $n_k$  satisfies (15.5.12). Then

$$\vartheta_k \leq \kappa_{\text{ubt}} \Delta_k^{1+\mu} \quad (15.5.16)$$

and

$$\vartheta(x_k + s_k) \leq \kappa_{\text{ubt}} \Delta_k^2. \quad (15.5.17)$$

for some constant<sup>264</sup>  $\kappa_{\text{ubt}} \geq 0$ .

**Proof.** If  $k \notin \mathcal{R}$ , we have from (15.5.12) and (15.5.5) that

$$\kappa_{\text{lsc}} \vartheta_k \leq \|n_k\| \leq \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}.$$

Now, the  $i$ th constraint function at  $x_k + s_k$  can be expressed as

$$c_i(x_k + s_k) = c_i(x_k) + \langle e_i, A_k s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} c_i(\xi_k) s_k \rangle$$

for  $i \in \mathcal{E} \cup \mathcal{I}$ , where we have used AW.1, the mean value theorem, and the fact that  $\xi_k$  belongs to the segment  $[x_k, x_k + s_k]$ . Using AI.1, we may bound the Hessian of the constraint functions and obtain from (15.5.1c), the Cauchy–Schwartz inequality, and (15.5.1b) that

$$|c_i(x_k + s_k)| \leq \frac{1}{2} \max_{x \in \mathcal{X}} \|\nabla_{xx} c_i(x)\| \|s_k\|^2 \leq \kappa_1 \Delta_k^2$$

if  $i \in \mathcal{E}$ , or

$$-c_i(x_k + s_k) \leq \frac{1}{2} \max_{x \in \mathcal{X}} \|\nabla_{xx} c_i(x)\| \|s_k\|^2 \leq \kappa_1 \Delta_k^2$$

if  $i \in \mathcal{I}$ , where we have defined

$$\kappa_1 \stackrel{\text{def}}{=} \frac{1}{2} \max_{i \in \mathcal{E} \cup \mathcal{I}} \max_{x \in \mathcal{X}} \|\nabla_{xx} c_i(x)\|.$$

This gives the desired bound with

$$\kappa_{\text{ubt}} = \max[\kappa_1, \kappa_\Delta \kappa_\mu / \kappa_{\text{lsc}}].$$

□

We next assess the model decrease when the trust-region radius is sufficiently small.

---

<sup>264</sup>“ubt” stands for “upper bound on theta”.

**Lemma 15.5.4** Suppose that AW.1, AA.1m, AI.1, and AM.4j hold, that  $k \notin \mathcal{R}$ , that

$$\chi_k \geq \epsilon, \quad (15.5.18)$$

and that

$$\Delta_k \leq \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \left( \frac{2\kappa_{\text{ubg}}}{\kappa_{\text{umh}}\kappa_\Delta\kappa_\mu} \right)^{\frac{1}{1+\mu}}, \left( \frac{\kappa_{\text{tmd}}\epsilon}{4\kappa_{\text{ubg}}\kappa_\Delta\kappa_\mu} \right)^{\frac{1}{\mu}} \right] \stackrel{\text{def}}{=} \delta_m, \quad (15.5.19)$$

where  $\kappa_{\text{ubg}} \stackrel{\text{def}}{=} \max_{x \in \mathcal{X}} \|\nabla_x f(x)\|$ . Then

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2}\kappa_{\text{tmd}}\epsilon\Delta_k.$$

**Proof.** We first note that, by AA.1m, AM.4j, (15.5.18), and (15.5.19),

$$m_k(x_k^N) - m_k(x_k + s_k) \geq \kappa_{\text{tmd}}\chi_k \min \left[ \frac{\chi_k}{\kappa_{\text{umh}}}, \Delta_k \right] \geq \kappa_{\text{tmd}}\epsilon\Delta_k. \quad (15.5.20)$$

Now

$$m_k(x_k^N) = m_k(x_k) + \langle g_k, n_k \rangle + \frac{1}{2}\langle n_k, H_k n_k \rangle$$

and therefore, using the Cauchy–Schwartz inequality, (15.5.5), and (15.5.19), we have that

$$\begin{aligned} |m_k(x_k) - m_k(x_k^N)| &\leq \|n_k\| \|g_k\| + \frac{1}{2}\|H_k\| \|n_k\|^2 \\ &\leq \kappa_{\text{ubg}}\|n_k\| + \frac{1}{2}\kappa_{\text{umh}}\|n_k\|^2 \\ &\leq \kappa_{\text{ubg}}\kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} + \frac{1}{2}\kappa_{\text{umh}}\kappa_\Delta^2\kappa_\mu^2\Delta_k^{2(1+\mu)} \\ &\leq 2\kappa_{\text{ubg}}\kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} \\ &\leq \frac{1}{2}\kappa_{\text{tmd}}\epsilon\Delta_k. \end{aligned}$$

We thus conclude from this last inequality and (15.5.20) that the desired conclusion holds.  $\square$

We continue our analysis by showing, as the reader has grown to expect, that iterations have to be very successful when the trust-region radius is sufficiently small.

**Lemma 15.5.5** Suppose that AW.1, AA.1m, AI.1, AM.4j, and (15.5.18) hold, that  $k \notin \mathcal{R}$ , and that

$$\Delta_k \leq \min \left[ \delta_m, \frac{(1-\eta_2)\kappa_{\text{tmd}}\epsilon}{2\kappa_{\text{ubh}}} \right] \stackrel{\text{def}}{=} \delta_\rho. \quad (15.5.21)$$

Then

$$\rho_k \geq \eta_2.$$

**Proof.** Using (15.5.9), (15.5.10), Lemma 15.5.4, and (15.5.21), we find that

$$|\rho_k - 1| \leq \frac{|f(x_k + s_k) - m_k(x_k + s_k)|}{|m_k(x_k) - m_k(x_k + s_k)|} \leq \frac{\kappa_{\text{ubh}} \Delta_k^2}{\frac{1}{2} \kappa_{\text{tmd}} \epsilon \Delta_k} \leq 1 - \eta_2,$$

from which the conclusion immediately follows.  $\square$

Now, we also show that the test (15.5.8) will always be satisfied in the above circumstances.

**Lemma 15.5.6** Suppose that AW.1, AA.1m, AI.1, AM.4j, and (15.5.18) hold, that  $k \notin \mathcal{R}$ , that  $n_k$  satisfies (15.5.12), and that

$$\Delta_k \leq \min \left[ \delta_m, \left( \frac{\kappa_{\text{tmd}} \epsilon}{2 \kappa_\vartheta \kappa_{\text{ubt}}^2} \right)^{\frac{1}{1+2\mu}} \right] \stackrel{\text{def}}{=} \delta_f. \quad (15.5.22)$$

Then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\vartheta \vartheta_k^2.$$

**Proof.** This directly results from the inequalities

$$\kappa_\vartheta \vartheta_k^2 \leq \kappa_\vartheta \kappa_{\text{ubt}}^2 \Delta_k^{2(1+\mu)} \leq \frac{1}{2} \kappa_{\text{tmd}} \epsilon \Delta_k \leq m_k(x_k) - m_k(x_k + s_k),$$

where we successively used Lemma 15.5.3, (15.5.22), and Lemma 15.5.4.  $\square$

We may also guarantee a decrease in the objective function, large enough to ensure that the trial point is acceptable with respect to the  $(\vartheta, f)$ -pair associated with  $x_k$ , so long as the constraint violation is itself sufficiently small.

**Lemma 15.5.7** Suppose that AW.1, AA.1m, AI.1, AM.4j, and (15.5.18) hold, that  $k \notin \mathcal{R}$ , that  $n_k$  satisfies (15.5.12), and that

$$\Delta_k \leq \min \left[ \frac{\eta_2 \kappa_{\text{tmd}} \epsilon}{2 \kappa_{\text{ubt}} \gamma_\vartheta}, \delta_\rho \right] \stackrel{\text{def}}{=} \delta_\vartheta. \quad (15.5.23)$$

Then

$$f(x_k + s_k) \leq f(x_k) - \gamma_\vartheta \vartheta(x_k + s_k).$$

**Proof.** Observe first that we may apply Lemmas 15.5.3–15.5.5 because of (15.5.18), (15.5.23), the hypothesis  $k \notin \mathcal{R}$ , and because  $n_k$  satisfies (15.5.12). As a consequence, (15.5.17) and (15.5.23) imply that

$$\vartheta(x_k + s_k) \leq \kappa_{\text{ubt}} \Delta_k^2 \leq \frac{\eta_2 \kappa_{\text{tmd}} \epsilon}{2 \gamma_\vartheta} \Delta_k.$$

Using this bound and Lemmas 15.5.4–15.5.5, we then obtain that

$$\begin{aligned} f(x_k) - f(x_k + s_k) &\geq \eta_2[m_k(x_k) - m_k(x_k + s_k)] \\ &\geq \frac{1}{2}\eta_2\kappa_{\text{tmd}}\epsilon\Delta_k \\ &\geq \gamma_\vartheta\vartheta(x_k + s_k), \end{aligned}$$

and the desired inequality follows.  $\square$

We now establish that if the trust-region radius and the constraint violation are both small at a noncritical iterate  $x_k$ , TRQP( $x_k, \Delta_k$ ) must be compatible.

**Lemma 15.5.8** Suppose that AW.1, AA.1l, AA.1m, AI.1, AM.4j, and (15.5.18) hold and that

$$\Delta_k \leq \min \left[ \gamma_0\delta_\vartheta, \left( \frac{1}{\kappa_\mu} \right)^{\frac{1}{\mu}}, \left( \frac{\gamma_0^2(1-\gamma_\vartheta)\kappa_\Delta\kappa_\mu}{\kappa_{\text{usc}}\kappa_{\text{ubt}}} \right)^{\frac{1}{1-\mu}} \right]. \quad (15.5.24)$$

Suppose furthermore that  $\vartheta_k$  is arbitrarily small. Then  $k \notin \mathcal{R}$ .

**Proof.** Because  $\vartheta_k$  is arbitrarily small, we know from AA.1l and Lemma 15.5.1 that  $n_k$  exists and satisfies (15.5.4) and (15.5.12). Assume, for the purpose of deriving a contradiction, that  $k \in \mathcal{R}$ , that is,

$$\|n_k\| > \kappa_\Delta\kappa_\mu\Delta_k^{1+\mu}, \quad (15.5.25)$$

where we have used (15.5.5) and (15.5.24). In this case, the mechanism of the algorithm ensures that  $k-1 \notin \mathcal{R}$ . Now assume that iteration  $k-1$  is unsuccessful. Because of Lemmas 15.5.5 and 15.5.7, which hold at iteration  $k-1 \notin \mathcal{R}$  because of (15.5.24), the fact that  $\vartheta_k = \vartheta_{k-1}$ , and AA.1l, we obtain that

$$\rho_{k-1} \geq \eta_2 \text{ and } f(x_{k-1} + s_{k-1}) \leq f(x_{k-1}) - \gamma_\vartheta\vartheta(x_{k-1} + s_{k-1}).$$

Hence, given that  $x_{k-1}$  is acceptable for the filter at the beginning of iteration  $k-1$ , if this iteration is unsuccessful, it must be because

$$\vartheta(x_{k-1} + s_{k-1}) > (1 - \gamma_\vartheta)\vartheta_{k-1} = (1 - \gamma_\vartheta)\vartheta_k.$$

But Lemma 15.5.3 and the mechanism of the algorithm then imply that

$$(1 - \gamma_\vartheta)\vartheta_k \leq \kappa_{\text{ubt}}\Delta_{k-1}^2 \leq \frac{\kappa_{\text{ubt}}}{\gamma_0^2}\Delta_k^2.$$

Combining this last bound with (15.5.25) and (15.5.4), we deduce that

$$\kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} < \|n_k\| \leq \kappa_{\text{usc}}\vartheta_k \leq \frac{\kappa_{\text{usc}}\kappa_{\text{ubt}}}{\gamma_0^2(1 - \gamma_\vartheta)}\Delta_k^2$$

and hence that

$$\Delta_k^{1-\mu} > \frac{\gamma_0^2(1 - \gamma_\vartheta)\kappa_\Delta\kappa_\mu}{\kappa_{\text{usc}}\kappa_{\text{ubt}}}.$$

Since this last inequality contradicts (15.5.24), our assumption that iteration  $k - 1$  is unsuccessful must be false. Thus iteration  $k - 1$  is successful and  $\vartheta_k = \vartheta(x_{k-1} + s_{k-1})$ . We then obtain from (15.5.25), (15.5.4), and (15.5.17) that

$$\kappa_\Delta \kappa_\mu \Delta_k^{1+\mu} < \|n_k\| \leq \kappa_{\text{usc}} \vartheta_k \leq \kappa_{\text{usc}} \kappa_{\text{ubt}} \Delta_{k-1}^2 \leq \frac{\kappa_{\text{usc}} \kappa_{\text{ubt}}}{\gamma_0^2} \Delta_k^2,$$

which is again impossible because of (15.5.24) and because  $(1 - \gamma_\vartheta) < 1$ . Hence our initial assumption (15.5.25) must be false, which yields the desired conclusion.  $\square$

We now distinguish two mutually exclusive cases. For the first, we consider what happens when an infinite subsequence of iterates is added to the filter.

**Lemma 15.5.9** Suppose that AW.1, AA.1l, AA.1m, AI.1, and AM.4j hold. Suppose furthermore that there exists an infinite subsequence  $\{k_j\} \in \mathcal{A}$ . Then we have that either the restoration procedure terminates unsuccessfully or

$$\lim_{j \rightarrow \infty} \vartheta_{k_j} = 0 \quad (15.5.26)$$

and

$$\lim_{j \rightarrow \infty} \chi_{k_j} = 0.$$

**Proof.** Suppose that the restoration procedure always terminates successfully. Let  $\{k_i\}$  be any infinite subsequence of  $\{k_j\}$ . We observe that (15.5.26) follows from Lemma 15.5.2. Combining this with AA.1l ensures that  $n_{k_i}$  exists and satisfies (15.5.4) for  $i \geq i_0$ , say, and therefore that

$$\lim_{i \rightarrow \infty} \|n_{k_i}\| = 0. \quad (15.5.27)$$

As we noted in the proof of Lemma 15.5.4,

$$|m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^N)| \leq \kappa_{\text{ubg}} \|n_{k_i}\| + \frac{1}{2} \kappa_{\text{umh}} \|n_{k_i}\|^2,$$

which in turn, with (15.5.27), yields that

$$\lim_{i \rightarrow \infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^N)] = 0. \quad (15.5.28)$$

Suppose now that

$$\chi_{k_i} \geq \epsilon_2 > 0 \quad (15.5.29)$$

for all  $i$  and some  $\epsilon_2 > 0$ . Suppose furthermore that there exists  $\epsilon_3 > 0$  such that, for all  $i \geq i_0$ ,

$$\Delta_{k_i} \geq \epsilon_3. \quad (15.5.30)$$

Then we deduce from AA.1m and AM.4j that

$$m_{k_i}(x_{k_i}^N) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{\text{tmd}} \epsilon_2 \min \left[ \frac{\epsilon_2}{\kappa_{\text{umh}}}, \epsilon_3 \right] \stackrel{\text{def}}{=} \delta > 0. \quad (15.5.31)$$

We now decompose the model decrease into its normal and tangential components, that is,

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) = m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^N) + m_{k_i}(x_{k_i}^N) - m_{k_i}(x_{k_i} + s_{k_i}).$$

Substituting (15.5.28) and (15.5.31) into this decomposition, we find that

$$\lim_{i \rightarrow \infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i})] \geq \delta > 0. \quad (15.5.32)$$

We now observe that, because  $x_{k_i}$  is added to the filter at iteration  $k_i$ , we know from the mechanism of the algorithm and from Lemma 15.5.8 that either iteration  $k_i \in \mathcal{R}$  or (15.5.8) must fail. If  $k_i \in \mathcal{R}$  for  $i \geq i_0$ , we obtain from the fact that (15.5.5) does not hold that

$$\|n_{k_i}\| > \kappa_\Delta \kappa_\mu \Delta_{k_i}^{1+\mu},$$

and (15.5.27) then implies that  $\Delta_{k_i}$  is arbitrarily small for  $i$  sufficiently large. This contradicts (15.5.30) and, hence, (15.5.8) must fail for  $i$  sufficiently large; that is,

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) < \kappa_\vartheta \vartheta_{k_i}^2. \quad (15.5.33)$$

Combining this bound with (15.5.32), we find that  $\vartheta_{k_i}$  is bounded away from zero for  $i$  sufficiently large, which is impossible in view of (15.5.26). We therefore deduce that (15.5.30) cannot hold and obtain that there is a subsequence  $\{k_\ell\} \subseteq \{k_i\}$  for which

$$\lim_{\ell \rightarrow \infty} \Delta_{k_\ell} = 0.$$

We now restrict our attention to the tail of this subsequence, that is, to the set of indices  $k_\ell$  that are large enough to ensure that (15.5.24) holds (with  $\epsilon$  replaced by  $\epsilon_2$ ) and that (15.5.23) also holds, which is possible because of (15.5.26). For these indices, we may therefore apply Lemma 15.5.8 and deduce that iteration  $k_\ell \notin \mathcal{R}$  for  $\ell$  sufficiently large. Hence, as above, (15.5.33) must hold for  $\ell$  sufficiently large. However, we may also apply Lemma 15.5.6, which contradicts (15.5.33), and therefore (15.5.29) cannot hold, yielding that

$$\liminf_{i \rightarrow \infty} \chi_{k_i} = 0.$$

The required result then follows since  $\{k_i\}$  is any infinite subsequence of  $\{k_j\}$ .  $\square$

Thus, if an infinite subsequence of iterates is added to the filter, this subsequence converges to a first-order critical point. Our remaining analysis then naturally concentrates on the possibility that there may be no such infinite subsequence. In this case, no further iterates are added to the filter for  $k$  sufficiently large. In particular, this means that the number of restoration iterations,  $|\mathcal{R}|$ , must be finite. In what follows, we assume that  $k_0 \geq 0$  is the last iteration for which  $x_{k_0-1}$  is added to the filter.

**Lemma 15.5.10** Suppose that AW.1, AA.1l, AA.1m, AI.1, and AM.4j hold. Suppose furthermore that (15.5.8) holds for all  $k \geq k_0$ . Then we have that

$$\lim_{k \rightarrow \infty} \vartheta_k = 0. \quad (15.5.34)$$

Furthermore,  $n_k$  exists and satisfies (15.5.4) and (15.5.12) for all  $k$  sufficiently large.

**Proof.** Consider any successful iterate with  $k \geq k_0$ . Then we have that

$$f(x_k) - f(x_{k+1}) \geq \eta_1 [m_k(x_k) - m_k(x_k + s_k)] \geq \eta_1 \kappa_\vartheta \vartheta_k^2 \geq 0. \quad (15.5.35)$$

Thus the objective function does not increase for all successful iterations with  $k \geq k_0$ . But AW.1 and AI.1 imply (15.5.11), and therefore we must have, from the first part of this statement, that

$$\lim_{k \rightarrow \infty} f(x_k) - f(x_{k+1}) = 0. \quad (15.5.36)$$

The limit (15.5.34) then follows from (15.5.35) and the fact that  $\vartheta_j = \vartheta_k$  for all unsuccessful iterations  $j$  that immediately follow the successful iteration  $k$ , if any. The last conclusion then results from AA.1l.  $\square$

We now show that the trust-region radius cannot become arbitrarily small if the (asymptotically feasible) iterates stay away from first-order critical points.

**Lemma 15.5.11** Suppose that AW.1, AA.1l, AA.1m, AI.1, and AM.4j hold. Suppose furthermore that (15.5.8) and (15.5.18) hold for all  $k \geq k_0$ . Then there exists a  $\Delta_{\min} > 0$  such that

$$\Delta_k \geq \Delta_{\min}$$

for all  $k$ .

**Proof.** Suppose that  $k_1 \geq k_0$  is chosen sufficiently large to ensure that (15.5.23) holds and that  $n_k$  exists and satisfies (15.5.4) for all  $k \geq k_1$ , which is possible because of Lemma 15.5.10. Suppose also, for the purpose of obtaining a contradiction, that iteration  $j$  is the first iteration following iteration  $k_1$  for which

$$\Delta_j \leq \gamma_0 \min \left[ \delta_\rho, \sqrt{\frac{(1 - \gamma_\vartheta) \vartheta^F}{\kappa_{\text{ubt}}}}, \Delta_{k_1} \right] \stackrel{\text{def}}{=} \gamma_0 \delta_s, \quad (15.5.37)$$

where

$$\vartheta^F \stackrel{\text{def}}{=} \min_{i \in \mathcal{A}} \vartheta_i$$

is the smallest constraint violation appearing in the filter. Note that the inequality  $\Delta_j \leq \gamma_0 \Delta_{k_1}$ , which is implied by (15.5.37), ensures that  $j \geq k_1 + 1$  and hence that  $j - 1 \geq k_1$ . Then the mechanism of the algorithm implies that

$$\Delta_{j-1} \leq \delta_s, \quad (15.5.38)$$

and Lemma 15.5.5, which is applicable with  $k$  replaced by  $j - 1$  because (15.5.37) and (15.5.38) together imply (15.5.21) and because  $j$  is sufficiently large to ensure that  $j - 1 \notin \mathcal{R}$ , then ensures that

$$\rho_{j-1} \geq \eta_2. \quad (15.5.39)$$

Furthermore, Lemma 15.5.3, (15.5.37), and (15.5.38) give that

$$\vartheta(x_{j-1} + s_{j-1}) \leq \kappa_{\text{ubt}} \Delta_{j-1}^2 \leq (1 - \gamma_\vartheta) \vartheta^F. \quad (15.5.40)$$

We may also apply Lemma 15.5.7 because (15.5.37) and (15.5.38) ensure that (15.5.21) holds and furthermore (15.5.23) also holds for  $j \geq k_1$ . Hence we deduce that

$$f(x_{j-1} + s_{j-1}) \leq f(x_{j-1}) - \gamma_\vartheta \vartheta(x_{j-1} + s_{j-1}).$$

This last relation and (15.5.40) ensure that  $x_{j-1} + s_{j-1}$  is acceptable for the filter. Combining this conclusion with (15.5.39), the fact that (15.5.8) holds for iteration  $j - 1$ , and the mechanism of the algorithm, we obtain that  $\Delta_j \geq \Delta_{j-1}$ . As a consequence, and since (15.5.8) also holds at iteration  $j - 1$ , iteration  $j$  cannot be the first iteration following  $k_1$  for which (15.5.37) holds. This contradiction shows that  $\Delta_k \geq \gamma_0 \delta_s$  for all  $k > k_1$ , and the desired result follows if we define

$$\Delta_{\min} = \min[\Delta_0, \dots, \Delta_{k_1}, \gamma_0 \delta_s]. \quad \square$$

We may now analyse the convergence of  $\chi_k$  itself.

**Lemma 15.5.12** Suppose that AW.1, AA.1l, AA.1m, AI.1, and AM.4j hold. Suppose furthermore that (15.5.8) holds for all  $k \geq k_0$ . Then

$$\liminf_{k \rightarrow \infty} \chi_k = 0.$$

**Proof.** We start by observing that Lemma 15.5.10 implies that (15.5.4) holds for  $k$  sufficiently large. Moreover, as in Lemma 15.5.10, we obtain (15.5.35) and therefore (15.5.36) for each  $k \in \mathcal{S}$ ,  $k \geq k_0$ . Suppose now, for the purpose of obtaining a contradiction, that (15.5.18) holds, and note that

$$m_k(x_k) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k^N) + m_k(x_k^N) - m_k(x_k + s_k). \quad (15.5.41)$$

Moreover, as in Lemma 15.5.4, that

$$|m_k(x_k) - m_k(x_k^N)| \leq \kappa_{\text{ubg}} \|n_k\| + \kappa_{\text{umh}} \|n_k\|^2,$$

which in turn yields that

$$\lim_{k \rightarrow \infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^N)] = 0$$

because of Lemma 15.5.10 and (15.5.4). This limit, together with (15.5.35), (15.5.36), and (15.5.41), then gives that

$$\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{S}}} [m_k(x_k^N) - m_k(x_k + s_k)] = 0. \quad (15.5.42)$$

But AA.1m, (15.5.18), AM.4j, and Lemma 15.5.11 together imply that, for all  $k \geq k_0$ ,

$$m_k(x_k^N) - m_k(x_k + s_k) \geq \kappa_{\text{tmd}} \chi_k \min \left[ \frac{\chi_k}{\beta_k}, \Delta_k \right] \geq \kappa_{\text{tmd}} \epsilon \min \left[ \frac{\epsilon}{\kappa_{\text{umh}}}, \Delta_{\min} \right], \quad (15.5.43)$$

immediately giving a contradiction with (15.5.42). Hence (15.5.18) cannot hold and the desired result follows.  $\square$

We may summarize all of the above in our main global convergence result.

**Theorem 15.5.13** Suppose that AW.1, AA.1l, AA.1m, AI.1, and AM.4j hold. Let  $\{x_k\}$  be a sequence of iterates produced by Algorithm 15.5.1. Then either the restoration procedure terminates unsuccessfully by converging to an infeasible first-order critical point of problem (15.5.6), or there is a subsequence  $\{k_j\}$  for which

$$\lim_{j \rightarrow \infty} x_{k_j} = x_*$$

and  $x_*$  is a first-order critical point for problem (15.1.1).

**Proof.** Suppose that the restoration iteration always terminates successfully. From Lemmas 15.5.9, 15.5.10, and 15.5.12, we obtain that, for some subsequence  $\{k_j\}$ ,

$$\lim_{j \rightarrow \infty} \vartheta_{k_j} = \lim_{j \rightarrow \infty} \chi_{k_j} = 0.$$

The convergence of  $\vartheta_{k_j}$  to zero and AA.1l then give that

$$\lim_{j \rightarrow \infty} \|n_{k_j}\| = 0,$$

and therefore that

$$\lim_{j \rightarrow \infty} \|x_{k_j}^N - x_{k_j}\| = 0.$$

The conclusion then follows from the continuity of both  $\vartheta$  and  $\chi$ .  $\square$

Can we dispense with AI.1 to obtain this result? Firstly, this assumption ensures that the objective and constraint functions remain bounded above and below (see (15.5.11)). This is crucial for the rest of the analysis because the convergence of the iterates to feasibility depends on the fact that the area of the filter is finite. Thus, if AI.1 does not hold, we have to verify that (15.5.11) holds for other reasons. The second part of this statement may be ensured quite simply by initializing the filter to  $(\vartheta^{\max}, -\infty)$ , for some  $\vartheta^{\max} > \vartheta_0$ , in Step 0 of the algorithm. This has the effect of putting an upper bound on the infeasibility of all iterates, which may be useful in practice. However, this does not prevent the objective function from being unbounded below in

$$\mathcal{C}(\vartheta^{\max}) = \{x \in \mathbb{R}^n \mid \vartheta(x) \leq \vartheta^{\max}\},$$

and we cannot exclude the possibility that a sequence of infeasible iterates might both continue to improve the value of the objective function and satisfy (15.5.8). If  $\mathcal{C}(\vartheta^{\max})$  is bounded, AI.1 is most certainly satisfied. If this is not the case, we could assume that

$$f^{\min} \leq f(x) \leq f^{\max} \text{ and } 0 \leq \vartheta(x) \leq \vartheta^{\max} \text{ for } x \in \mathcal{C}(\vartheta^{\max}) \quad (15.5.44)$$

for some values of  $f^{\min}$  and  $f^{\max}$  and simply monitor that the values  $f(x_k)$  are reasonable, in view of the problem being solved, as the algorithm proceeds. In the light of AW.1, AI.1 also ensures the boundedness of first and second derivatives. This is perhaps less crucial, as this may well remain true even if AI.1 does not hold.

To summarize, we may replace AI.1 and AW.1 by the following assumption.

**AW.1g** The functions  $f$  and  $c$  are twice-continuously differentiable on an open set containing  $\mathcal{C}(\vartheta^{\max})$ , (15.5.44) holds, and their first and second derivatives are uniformly bounded over  $\mathcal{C}(\vartheta^{\max})$ .

The reader should note that the comments following the statement of AA.11 no longer apply if limit points at infinity are allowed. This means that AW.1g is actually slightly stronger than the combination of AA.11 and AI.1.

### 15.5.5 An Alternative Step Strategy

It is also interesting to return to the question of whether it is possible to find a cheaper alternative to computing a normal step, finding a generalized Cauchy point, and explicitly checking AA.1m. Suppose, for now, that it is possible to compute a point  $x_k + s'_k$  directly to satisfy the constraints of TRQP( $x_k, \Delta_k$ ) and for which

$$m_k(x_k) - m_k(x_k + s'_k) \geq \epsilon_1 \min[\pi_k, \Delta_k] \quad (15.5.45)$$

for a small positive constant  $\epsilon_1$  and  $\pi_k = \pi(x_k)$ , where  $\pi$  is a continuous function of its argument. Furthermore, consider the following algorithm.

**Algorithm 15.5.2: Single-step SQP filter algorithm**

Same as Algorithm 15.5.1 (with  $s_k$  replaced by  $s'_k$ ), except that Steps 1 and 2 are replaced by the following.

**Step 1: Ensure compatibility.** If the feasible set of  $\text{TRQP}(x_k, \Delta_k)$  is empty, include  $x_k$  in the filter and compute a restoration step  $r_k$  such that  $\text{TRQP}(x_k + r_k, \Delta_{k+1})$  is compatible for some  $\Delta_{k+1} > 0$  and  $x_k + r_k$  is acceptable for the filter. If this proves impossible, stop. Otherwise, define  $x_{k+1}$  to be  $x_k + r_k$  and go to Step 6.

**Step 2: Determine a trial step.** Compute a step  $s'_k$  that is feasible for this subproblem and such that (15.5.45) holds.

Interestingly, most of the properties of Algorithm 15.5.1 remain true for the modified Algorithm 15.5.2, as we shall now see by reconsidering the convergence theory of the previous section. Lemmas 15.5.1 and 15.5.2 are unmodified. Replacing  $n_k$  by  $s'_k$  in Lemma 15.5.3 yields (15.5.16) with  $\mu = 0$ —if  $s'_k$  can be computed, this implies that we do not have to worry about the existence of  $n_k$ . As the proof of (15.5.17) is also unmodified, we conclude that Lemma 15.5.3 remains true with  $\mu = 0$ . We now suppose, instead of (15.5.18), that

$$\pi_k \geq \epsilon$$

and obtain the conclusion of Lemma 15.5.4 immediately from (15.5.45). Lemma 15.5.5 is again unmodified, while Lemma 15.5.6 remains true if one uses the modified version of Lemma 15.5.3 (with  $\mu = 0$ ) to deduce its conclusion. The same applies to Lemma 15.5.7. The proof of Lemma 15.5.8 remains true but needs a slightly less trivial modification. In particular, the bound in (15.5.24) needs to be understood with the particular values

$$\mu = 0 \text{ and } \kappa_\Delta = \kappa_\mu = 1. \quad (15.5.46)$$

The existence of  $n_k$  at the beginning of the proof is a direct result of the existence of  $s'_k$ . The rest of the proof follows immediately using (15.5.46). We may dispense with the first part of the proof of Lemma 15.5.9 and immediately assume that

$$\pi_{k_i} \geq \epsilon_2 \quad (15.5.47)$$

(instead of (15.5.29)). We may then continue the proof of this lemma as stated using (15.5.46), except that we do not have to consider the decomposition of the step into tangential and normal components to obtain (15.5.32) from (15.5.31), and that we should substitute  $s'_{k_i}$  for  $n_{k_i}$  when necessary. Lemma 15.5.10 and 15.5.11 are unmodified. The proof of Lemma 15.5.12 simplifies because we may deduce (15.5.43) directly from (15.5.47) (instead of (15.5.29)) and (15.5.45). Gathering those results, we may therefore deduce the following result.

**Theorem 15.5.14** Suppose that AW.1, AM.4js, and AI.1 hold. Let  $\{x_k\}$  be a sequence of iterates produced by Algorithm 15.5.2. Then either the restoration procedure terminates unsuccessfully by converging to an infeasible first-order critical point of problem (15.5.6), or there is a subsequence  $\{k_j\}$  for which

$$\lim_{j \rightarrow \infty} \vartheta_{k_j} = 0$$

and

$$\lim_{j \rightarrow \infty} \pi_{k_j} = 0.$$

Thus we have established convergence of a subsequence to a first-order critical point provided we can exhibit a continuous function  $\pi(x)$  such that

$$\pi(x_*) = \vartheta(x_*) = 0$$

is equivalent to the first-order criticality of  $x_*$  and such that a step satisfying (15.5.45) can always be found. For instance, we might consider

$$\pi(x) = \|g(x) - A(x)^T y^{\text{LS}}(x)\|,$$

where  $y^{\text{LS}}(x)$  is the vector of least-squares multipliers at  $x$ . We may even replace this vector at iteration  $k$  by any  $y_k$  provided we can guarantee that  $\|y_k - y_k^{\text{LS}}\|$  converges to zero when  $\vartheta_k$  converges to zero. Of course, the main difficulty is still to find a step  $s'_k$  satisfying (15.5.45). If we are unable to find such a step for a particular iteration, we can always return to the composite-step technique we considered in the previous section. This then results in a hybrid but potentially efficient algorithm.

It is interesting to note that the role of the filter has been limited to ensuring that the iterates that are included in it converge to feasible points (see Lemma 15.5.2). This is achieved by making sure that each point in the filter reduces the surface of the “available” filter area and by setting a small margin around every infeasible point in the filter in order to prevent the convergence of a subsequence of iterates to such a point. Other strategies are possible that have the same effect. For instance, it is possible to only exclude from the filter area a small neighbourhood of infeasible points whose diameter is proportional to the constraint violation.

We conclude our introduction to filter methods by remarking that the result of Theorem 15.5.14 provides a conceptual equivalent, in the context of constrained optimization, of the convergence results for general criticality measures discussed in Chapter 8.

## Notes and References for Section 15.5

The idea of using a filter as an alternative to penalty functions—albeit using a number of heuristics not discussed here—was introduced by Fletcher and Leyffer (1997). The resulting algorithm forms the basis of the package filterSQP (see Fletcher and Leyffer, 1998), which

has been shown to be very efficient on a large set of test problems. The first step towards a formal convergence theory was given by Fletcher, Leyffer, and Toint (1998), who assumed a linear rather than a quadratic model (see Section 15.6), resulting in an SLP algorithm. Our exposition for the SQP case is based on Fletcher et al. (1999) with improvements borrowed from Chin and Fletcher (1999). A suitable numerical algorithm for the restoration procedure is described by Fletcher and Leyffer (1997), where the  $\ell_1$  norm is used to measure constraint violation. Trust-region procedures for the  $\ell_2$  norm are analysed by El-Hallabi and Tapia (1995) and Dennis, El-Alem, and Williamson (1999). See Section 16.2 for further details.

While initial numerical experiments suggest that the filter algorithm may be one of the most effective SQP methods, the concept is probably too new for a true assessment. Further developments seem likely. On the theoretical side, second-order and local convergence analyses (paralleling those for the other methods we have considered in this chapter) remain to be developed. It is not difficult to anticipate that these will rely on specific updating rules for the Lagrange multipliers, second-order corrections, and possibly the use of a modified definition of  $\rho_k$  given by

$$\rho_k = \frac{f(x_k) - f(x_k + s_k) + \frac{1}{2}\langle s_k, (H_k - B_k)s_k \rangle}{m_k(x_k) - m_k(x_k + s_k)}$$

(as an alternative to (15.5.9)). Another interesting possibility is to relax the conditions on the normal step  $n_k$  by requiring that  $x_k + n_k$  satisfies the constraints of  $QP(x_k)$  only approximately. This would possibly reduce the number of restoration iterations and allow a corresponding relaxation on the requirement that the tangential step has to maintain feasibility for the linearized constraints. On the more practical side, further numerical comparisons will be necessary before the seemingly high potential of this new class of methods can be truly confirmed. In particular, algorithms based on further relaxations of the filter concept, such as the one suggested in Section 15.5, have to be practically evaluated. The most efficient mix between the two step strategies described above also needs further investigation.

An interesting alternative to filter methods for equality-constrained problems has been very recently proposed by Ulbrich and Ulbrich (1999), where nonmonotone independent acceptance rules are used for the normal and tangential parts of the SQP step. As in the filter method, this approach ensures global convergence while avoiding the use of a penalty function.

## 15.6 Nonquadratic Models

We have concentrated in this chapter on second-order models, chiefly because of their potential for fast asymptotic convergence. From the perspective of global convergence, a first-order model is equally good, and all of the methods we have considered are applicable with Hessian approximations  $H_k = 0$ . From a computational perspective, the absence of curvature is an advantage since the subproblems are, or often may be reformulated as, linear programs. As the reader is undoubtedly aware, there are many excellent linear programming codes, and these may be immediately used to good effect. Although we would not usually recommend these *sequential linear programming* (SLP) methods when close to a critical point of the underlying problem, they are often quite effective when the problem is mildly nonlinear or when moving from a distant starting value towards the asymptotic region. They are sometimes used in the latter case as a first phase, from which a switch to a faster (second-order) method is made

as soon as a suitable neighbourhood of a critical point is detected. Unfortunately, in practice, determining such a region is often rather difficult. In some very large cases, SLP methods may be the only possible option, since it is fair to say that in general linear programming codes are currently able to solve significantly larger problems than are practicable for modern quadratic programming algorithms.

While a linear model offers considerable simplifications, it may, perversely, sometimes be worth complicating the model. In particular, the reader might have wondered why we have considered a quadratic model of the objective but only linear approximations to the constraints. After all, one might have expected quite reasonably to have used quadratic approximations throughout. The reason is purely pragmatic: minimizing a (possibly nonconvex) quadratic function subject to a set of (possibly nonconvex) quadratic constraints—this is commonly known as a *quadratically constrained* quadratic program (QCQP)—is usually a far harder problem than solving a quadratic program. There may, however, be some advantage in using quadratic constraints. Firstly, when the true constraints are highly nonlinear, a linear approximation may only be valid within a very small trust region, and convergence may be slow. Quadratic approximations may enlarge the region of applicability. Secondly, the feasible region for a quadratic model of the constraints can be larger than for a linear model, and the consequent trust-region subproblem may remain compatible with a smaller trust region. Thirdly, there is no longer any need to mix the curvature of the constraints with that of the objective in the Hessian of the model of the latter. And finally, Lagrange multiplier estimates generated by solving an appropriate QCQP are often more accurate than their quadratic programming counterparts. For these reasons, it is tempting to believe that such methods will become more widely available in the future. At present, such techniques are, unfortunately, largely impractical.

## Notes and References for Section 15.6

See Fletcher and Sainz de la Maza (1989), Zhang (1989), Ferris and Zavriev (1996), and Fletcher, Leyffer, and Toint (1998) for a variety of suitable SLP methods.

To our knowledge, the first practical method to use quadratic approximations to the constraints is the linesearch-based approach of Maany (1987). It is only relatively recently that trust-region methods of this sort have been considered (see Psiaki and Park, 1995; Kruk and Wolkowicz, 1998). Efficient methods for solving the QCQP subproblem when all the models are convex have been proposed by Phan huy Hao (1982), Goldfarb, Liu, and Wang (1991), Jarre (1991), Mehrotra and Sun (1991), and Nemirovskii and Scheinberg (1996). We are not aware of any special methods appropriate for general nonconvex QCQPs, although Anstreicher et al. (1999) hint that suitable methods might yet be possible for some important special cases.

## 15.7 Concluding Remarks

It would be fair to say that currently most of the exciting developments in nonlinear programming are in algorithms for nonlinearly constrained optimization. If we were

writing this five, or perhaps even two, years ago, we might have been slightly cautious about the applicability of SQP methods for large-scale computation, even though their marked superiority for small problems had been long recognized. However, at last we have started to see the development and comparison of SQP methods capable of coping effectively with (what one might call) medium-sized problems (that is, by today's standards, problems involving thousands or perhaps tens of thousands of unknowns<sup>265</sup>). This upturn in fortune for SQP methods has arisen for a number of reasons.

Firstly, and rather trivially, computer memory and CPU performance has improved tremendously over the past few years. Secondly, so has the ability to solve linear systems of equations—either directly or, with the aid of sophisticated preconditioners, iteratively. But more profoundly from our point of view, the interior-point revolution is having a dramatic effect on the number of systems of equations that need to be solved, since the total number of iterations required for such methods is often extremely modest<sup>266</sup> in comparison with earlier active set methods. Finally, we are now developing ways of truncating the solution of subproblems—just as we have done successfully for many years in the unconstrained case—so that increasingly accurate solutions are only found when they are needed, normally in the asymptotics.

In this chapter, we have described a number of practical SQP trust-region methods. It is only fair to the reader that we should try to compare and contrast them, although we do so knowing that future developments may quickly invalidate such conclusions. As there are currently a number of projects in progress to compare many of the methods we have considered here on large suites of test problems, we expect to understand the relative behaviour far better in the next few years.

We have covered five basic trust-region SQP methods:

- (1) those based on minimizing a model of a nonsmooth penalty function, as pioneered by Fletcher (Section 15.3.2);
- (2) those like Vardi's, which relax the linearized constraints (Section 15.4.1);
- (3) those related to Byrd and Omojokun's proposal, in which the linearized constraints are satisfied as closely as possible (Section 15.4.2);
- (4) those like Celis, Dennis, and Tapia's, which replace the linearized constraints by a single constraint on the norm of their infeasibility (Section 15.4.3); and
- (5) those that try to do away with a merit function altogether, due initially to Fletcher and Leyffer (Section 15.5).

We feel that the weakest of these currently are those in category (4), simply because the model problem that must be solved at each stage appears to be intractable, while

---

<sup>265</sup>This is small by comparison with the linear programs in up to 10 million unknowns that have now been solved.

<sup>266</sup>We should be slightly cautious here, since the problems we are most interested in are frequently nonconvex, and there is certainly no assurance that the guaranteed excellent performance of interior-point methods for convex problems will carry over to these harder problems. However, accumulating evidence is starting to show exciting gains in performance even in these cases.

the theoretical guarantees are no stronger than its competitors'. We were pleasantly surprised, however, with the methods in category (2), particularly since these methods have attracted a fair amount of dismissive criticism in the literature.<sup>267</sup> It is true that such methods depend on a number of parameters—we believe that this may be said of all of the competition, whatever is claimed elsewhere—but we have not been persuaded that these parameters are any more (or less) easy to select than for other algorithms. The main defect with the methods in category (2) is that they require that the linearized constraints be compatible at each stage, which is unrealistic in practice. The superficially-very-similar methods in category (3) avoid this disadvantage by aiming instead for a “least-incompatible” point at each stage. The algebraic requirements for methods in both categories are very similar, and given this, we feel that the methods in category (3) have a slight edge.

It remains to be seen which of the three leading contenders, those in categories (1), (3), and (5), proves to be the best. We suspect that there will be no overall winner. Those in categories (3) and (5) have an advantage at present because, by separating the computation of the step into two well-understood parts, it is easy to appeal to existing theory to decide how accurately each part should be solved. For methods in category (1), it appears that a quadratic program needs to be solved at each step, and while we can define a suitable Cauchy step for the model, it is not known how much better than this Cauchy step we need to do to obtain good practical performance. However, one clear advantage of methods in this category is that the initial point is feasible for each quadratic programming subproblem, and thus progress on both linearized constraint infeasibility and optimality may be made as the step proceeds. Moreover, such methods operate under perhaps the weakest of all the sets of assumptions of any method in this chapter.

We believe that the methods in category (5) are the most exciting development in SQP methods for many years. The idea of letting the basic SQP method run with as little interference as possible is most appealing, and the filter idea appears to be an effective way of achieving this, even though the convergence theory we give here requires more interference than perhaps we would have liked. Since this is a snapshot of an evolving idea, we would not be surprised if major advances were made for methods in this category in the near future.

---

<sup>267</sup>See, for example, Boggs and Tolle (1995), Coleman and Yuan (1995), and Dennis, El-Alem, and Maciel (1997).

# Chapter 16

---

## Nonlinear Equations and Nonlinear Fitting

---

### 16.1 Nonlinear Equations, Nonlinear Least Squares

We now consider a class of problems that is conceptually close to both constrained and unconstrained minimization: the solution of sets of nonlinear equations. The problem is to find a point  $x$  of  $\mathbb{R}^n$  such that

$$c(x) = 0, \quad (16.1.1)$$

where we assume here that  $c$  is a twice-continuously differentiable function from  $\mathbb{R}^n$  into  $\mathbb{R}^m$ . For any given point  $x$ , we call  $c(x)$  the *residual* of equation (16.1.1). If  $m = n$ , we obtain a square system of nonlinear equations. On the one hand, the problem can be viewed as a constrained problem without an objective function. On the other, it can be interpreted as a generalization of the first-order optimality conditions for an unconstrained problem, in the sense that we do not suppose that  $c(x)$  is the gradient of any function and thus that its derivatives are symmetric. If  $m > n$  (or if (16.1.1) does not have a solution), we may wish to find an  $x$  such that (16.1.1) holds “as closely as possible”. This means that, in all cases, our aim is to reduce  $\|c(x)\|_2$  as much as we can—if possible to zero. Our problem can then be rewritten as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \|c(x)\|_2^2, \quad (16.1.2)$$

where squaring the norm has the advantage of making the objective function of (16.1.2) everywhere twice-continuously differentiable.

One of the most important applications is *nonlinear fitting*, in which a model  $h(x, z)$  of some process, depending, possibly nonlinearly, on  $n$  unknown parameters  $x$  and input  $z$ , is required to reproduce the output  $y$  for a number of input-output pairs  $\{(z_i, y_i)\}_{i=1}^m$ . For instance, one may wish to estimate the parameters  $x$  of a biological model for brain cells from positron camera measurements ( $y_i$ ) of a radioactive tracer

injected in a patient's blood at a known rate ( $z_i$ ) and subsequently reaching the brain cells (the camera is placed close to the patient's head).<sup>268</sup> In such situations, one is led to a minimization problem of the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^m [h(x, z_i) - y_i]^2. \quad (16.1.3)$$

This is, of course, nothing other than problem (16.1.2), where the  $i$ th component of the vector  $c(x)$  is given by

$$c_i(x) = h(x, z_i) - y_i \quad (i = 1, \dots, m).$$

It is often the case that  $m$  is significantly larger than  $n$  and that experimental errors are such that  $h(x, z_i) \neq y_i$  even with the best possible  $x$ . When the errors are normally distributed, fitting in the  $\ell_2$  norm has some statistical significance since it minimizes the variance.

Problem (16.1.2) is called the *nonlinear least-squares problem* by reference to this squared  $\ell_2$  norm. It is important to realize at this point that the use of other norms is possible and sometimes preferable, as we will discuss in the next section. Assuming for now that we are content with the least-squares formulation (16.1.2), we could simply define

$$f(x) = \frac{1}{2} \|c(x)\|_2^2 \quad (16.1.4)$$

and apply Algorithm BTR (Algorithm 6.1.1, in which Step 2 would then correspond to finding a step that sufficiently reduces a model of the norm of the residual of the equation (16.1.1)) to problem (16.1.2). Note that the value of the solution of problem (16.1.1) will be nonzero in the important case where the system  $c(x) = 0$  does not have a solution. The theory of Sections 6.2 to 6.6 would then give suitable convergence results under assumptions AF.1–AF.3 on  $f$ , and we could stop the discussion at this point.

However, there are some interesting additional observations that apply to the nonlinear least-squares problem. The first is that assuming properties of  $f$  is not really desirable, since the real data for our problem is specified by the function  $c$ . We therefore prefer to express our assumptions as assumptions on this latter function. Interestingly, the assumptions on  $c$  are slightly different from those on  $f$ .

**AC.1e** The function  $c(x)$  is twice-continuously differentiable on  $\mathbb{R}^n$ .

**AC.3e** There exists a positive constant<sup>269</sup>  $\kappa_{\text{udc}}$  such that

$$\|c(x)\|_2 \leq \kappa_{\text{udc}}, \quad \|A(x)\|_2 \leq \kappa_{\text{udc}}, \quad \text{and} \quad \|\nabla_{xx} c_i(x)\|_2 \leq \kappa_{\text{udc}}$$

for all  $x \in \mathbb{R}^n$  and every  $i = 1, \dots, m$ , where  $A(x)$  is the Jacobian matrix of  $c$  at  $x$ .

<sup>268</sup>This application was brought to our attention by N. Dautrebande from the Université Catholique de Louvain, Belgium, and is embodied in the problems BRAINPC0 to BRAINPC9 in the CUTE collection of test problems.

<sup>269</sup>“udc” stands for “upper bound on the derivatives (including the zeroth) of the components of  $c$ ”.

Of course, these assumptions are sufficient for us to recover our familiar assumptions of  $f(x)$ , as we now show.

**Theorem 16.1.1** Suppose that AC.1e and AC.3e hold. Then AF.1–AF.3 hold for

$$f(x) = \frac{1}{2} \|c(x)\|_2^2.$$

**Proof.** The twice differentiable nature of the square of the norm  $\|\cdot\|_2$  and AC.1e immediately ensure AF.1. The fact that  $\|c(x)\|_2$  is obviously nonnegative for all  $x$  yields AF.2. Finally, the identity

$$\nabla_{xx} f(x) = A(x)^T A(x) + \sum_{i=1}^m c_i(x) \nabla_{xx} c_i(x) \quad (16.1.5)$$

and AC.3e guarantee AF.3.  $\square$

Observe that we do not have to assume any lower bound on the functions involved in the problem, since a lower bound is automatically supplied by the use of the norm. On the other hand, as is obvious, the boundedness of the Hessian of  $f$  requires the boundedness not only of the Hessians but also of the components of  $c$  and their gradients.

The second interesting aspect of trust-region methods applied to systems of nonlinear equations is that we may choose to model  $c(x)$  instead of  $f(x)$ . Let us denote the model of  $c(x_k + s)$  by  $m_k^c(x_k + s)$ , which is now a vector of  $\mathbb{R}^m$  whose  $i$ th component models the  $i$ th component of the vector  $c(x)$ . It is then quite natural to use the squared norm of this model, that is,

$$m_k^f(x_k + s) \stackrel{\text{def}}{=} \frac{1}{2} \|m_k^c(x_k + s)\|_2^2,$$

as a model of the objective function (16.1.4) of our associated optimization problem. Of course, the accuracy of this overall model depends on that of  $m_k^c$ , but in a very interesting way.

Indeed, consider a first-order model of the form

$$m_k^c(x_k + s) = m_k^c(x_k) + A_k s, \quad (16.1.6)$$

where  $A_k$  is some  $m \times n$  matrix. Then

$$m_k^f(x_k + s) = \frac{1}{2} \langle m_k^c(x_k) + A_k s, m_k^c(x_k) + A_k s \rangle = \frac{1}{2} \|m_k^c(x_k)\|_2^2 + \langle m_k^c(x_k), A_k s \rangle + \frac{1}{2} \|A_k s\|_2^2,$$

and we obtain a second-order model for  $f(x_k + s)$ ! The model (16.1.6) is called the *Gauss–Newton model* and is extremely popular in practice, mostly because the trust-region subproblem reduces to a linear least-squares calculation when the trust-region radius is large enough. However, as the Gauss–Newton model is always convex, and

since we saw in Section 6.5 that there is a possibility that convex models may force convergence of Algorithm BTR to critical points that are not second-order critical, we may prefer to include full second-order information. This can be done by considering

$$m_k^f(x_k + s) = m_k^f(x_k) + \langle A_k^T m_k^c(x_k), s \rangle + \frac{1}{2} \|A_k s\|_2^2 + \frac{1}{2} \sum_{i=1}^m m_{ik}^c(x_k) \langle s, \nabla_{xx} m_{ik}^c(x_k) s \rangle, \quad (16.1.7)$$

where  $m_{ik}^c(x_k + s)$  is the  $i$ th component of  $m_k^c(x_k + s)$ . This second-order model of  $f$  is known as the *Newton model* and is directly inspired by (16.1.5). If instead we consider a second-order model of  $c$ , such as

$$m_k^c(x_k + s) = m_k^c(x_k) + A_k s + \frac{1}{2} \begin{pmatrix} \langle s, \nabla_{xx} m_{1k}^c(x_k) s \rangle \\ \vdots \\ \langle s, \nabla_{xx} m_{mk}^c(x_k) s \rangle \end{pmatrix}, \quad (16.1.8)$$

we will see that this provides yet more information than (16.1.7). To simplify notation, we will denote

$$w_k(u, v) \stackrel{\text{def}}{=} \begin{pmatrix} \langle u, \nabla_{xx} m_{1k}^c(x_k) v \rangle \\ \vdots \\ \langle u, \nabla_{xx} m_{mk}^c(x_k) v \rangle \end{pmatrix}.$$

Considering (16.1.8), we deduce that

$$\begin{aligned} m_k^f(x_k + s) &= \frac{1}{2} \|m_k^c(x_k + s)\|_2^2 \\ &= \frac{1}{2} \|m_k^c(x_k)\|_2^2 + \langle m_k^c(x_k), A_k s \rangle + \frac{1}{2} \|A_k s\|_2^2 \\ &\quad + \frac{1}{2} \langle m_k^c(x_k), w_k(s, s) \rangle + \frac{1}{2} \langle A_k s, w_k(s, s) \rangle + \frac{1}{8} \|w_k(s, s)\|_2^2 \\ &= \frac{1}{2} \|m_k^c(x_k)\|_2^2 + \langle m_k^c(x_k), A_k s \rangle + \frac{1}{2} \|A_k s\|_2^2 \\ &\quad + \frac{1}{2} \sum_{i=1}^m m_{ik}^c(x_k) \langle s, \nabla_{xx} m_{ik}^c(x_k) s \rangle + \frac{1}{2} \langle A_k s, w_k(s, s) \rangle + \frac{1}{8} \|w_k(s, s)\|_2^2, \end{aligned}$$

and we see contributions from third- and fourth-order terms in the value of the model  $m_k^f$  at  $x_k + s$ . Again, the model  $m_k^f$  is of an order higher than  $m_k^c$ .

For the models (16.1.6), (16.1.7), or (16.1.8) or any further generalizations, we still have to clarify the assumptions we shall impose. These are as follows.

**AM.1e** The model  $m_k^c$  is twice-continuously differentiable on  $\mathcal{B}_k$ .

**AM.2e** The model coincides with  $c$  at  $x_k$ ; that is, for all  $k$ ,

$$m_k^c(x_k) = c(x_k).$$

**AM.3e** The first derivative of the model coincides with that of  $c$  at  $x_k$ ; that is,

$$A_k = A(x_k)$$

for all  $k$ .

**AM.4e** The Hessians of each of the components of the model are uniformly bounded within the trust region; that is,

$$\|\nabla_{xx} m_{ik}^c(x)\|_2 \leq \kappa_{\text{umh}} - 1$$

for all  $k$ , for all  $x \in \mathcal{B}_k$  and for every  $i = 1, \dots, m$ , where  $m_{ik}^c(x)$  is the  $i$ th component of  $m_k^c(x)$ .

**AM.5e** For  $i = 1, \dots, m$ ,

$$\lim_{k \rightarrow \infty} \|\nabla_{xx} c_i(x_k) - \nabla_{xx} m_{ik}^c(x_k)\|_2 = 0 \quad \text{when} \quad \lim_{k \rightarrow \infty} \|A_k^T m_{ik}^c(x_k)\|_2 = 0.$$

Algorithm BTR may be applied using any of these models and the convergence results of Sections 6.2 to 6.6 still apply, provided we verify that AM.1e–AM.4e (on  $m_k^c$ ) imply AM.1–AM.4 (on  $m_k$ ). This is the object of the following easy proposition.

**Theorem 16.1.2** Suppose AC.3e and AM.1e–AM.4e hold. Then AM.1–AM.4 hold for the model  $m_k^f = \frac{1}{2}\|m_k^c\|_2^2$ . Moreover, if AM.5e holds, then AM.5 holds for  $m_k^f$ .

**Proof.** AM.1 immediately results from AM.1e and the twice-continuously differentiable nature of the square of the norm  $\|\cdot\|_2$ . AM.2e yields AM.2. Moreover, AM.3 is a consequence of the identity

$$\nabla_x m_k^c(x_k) = A_k^T m_k^c(x_k) = A(x_k)^T c(x_k),$$

AM.2e, and AM.3e. AM.4 follows from (16.1.5), AC.3e, AM.3e, and AM.4e. Finally, AM.5 is a direct consequence of (16.1.5), AM.2e, AM.3e, and AM.5e.  $\square$

Observe here that AM.5e and thus the second conclusion of Theorem 16.1.2 do not necessarily hold<sup>270</sup> if the Gauss–Newton model is used, because the Hessian of this model lacks the terms depending on the Hessians of the individual components of  $c$ .

Some words of warning are necessary at this point. Although the theory of Sections 6.2 to 6.6 ensures convergence of the iterates produced by Algorithm BTR to first-order or second-order critical points, this does not imply that any of these points actually satisfies  $f(x) = 0$ . Indeed, this would mean that this limit point has a zero residual and is therefore a global minimizer of  $f(x)$ . Figure 16.1.1 shows the curves corresponding to the two-dimensional system of nonlinear equations

$$c(x_1, x_2) = \begin{pmatrix} 1 + \frac{7}{10}x_1 - \frac{2}{2+\cos(2x_1)} - x_2 \\ x_1(x_1 - 1)(x_1 + 1) - 10x_2 \end{pmatrix} = 0$$

<sup>270</sup>Although, as we will see below, it does hold in the case when  $\lim_{k \rightarrow \infty} c(x_k) = 0$ .

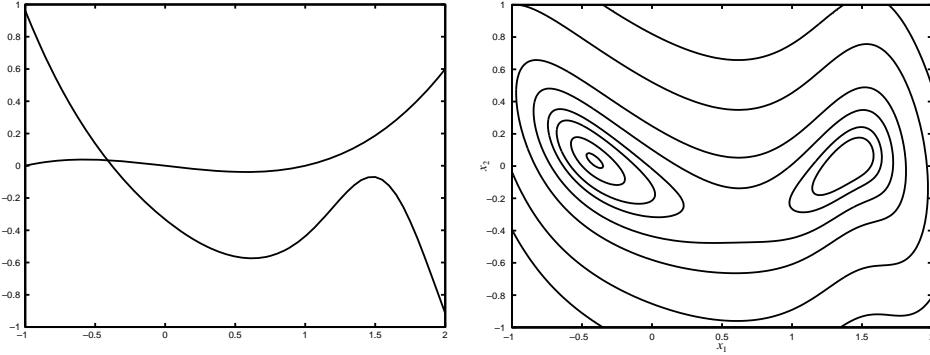


Figure 16.1.1: A system of nonlinear equations in two variables and the contour lines of the corresponding squared residual norm.

(on the left) and the contour lines of the associated residual squared norm (on the right). One immediately sees in this figure that not every minimum of the squared residual norm corresponds to a zero: the rightmost minimum in the right picture corresponds to a point where the curves of the left picture approach each other without intersecting.

The first reason why  $f(x)$  might be nonzero is that, unfortunately, Algorithm BTR does not guarantee finding global minimizers. The second is that there may not be any roots, particular when  $m > n$ . One should note here that these difficulties are not specific to Algorithm BTR or to trust-region methods in general: if there is no root, no algorithm can possibly find one. And, if there is actually a root, finding it by minimizing any measure of the size of the residual involves finding a global minimizer, which is in general a very difficult task, especially when  $n$  is large.

However, despite these warnings, roots are often found in practice—usually because a good starting point is available—and fortunately it frequently happens that a limit point  $x_*$  of the iterates produced by Algorithm BTR is a root of  $c$ . Assuming this desirable situation occurs, we may make two further observations.

Firstly, we see that the Gauss–Newton model is asymptotically a full second-order model when approaching a root.

**Theorem 16.1.3** Suppose that AC.1e, AC.3e, and AM.1e–AM.4e hold. Suppose furthermore that  $\{x_{k_i}\}$  is a subsequence of iterates generated by Algorithm BTR converging to  $x_*$  for which  $c(x_*) = 0$ . Suppose finally that Algorithm BTR uses the Gauss–Newton model (16.1.6). Then

$$\lim_{i \rightarrow \infty} \|\nabla_{xx} f(x_{k_i}) - \nabla_{xx} m_k^f(x_{k_i})\|_2 = 0 \quad (16.1.9)$$

and thus AM.5 holds.

**Proof.** The limit (16.1.9) results immediately from the observation that the dif-

ference between the Hessians of  $f$  and the model is

$$\nabla_{xx} f(x_{k_i}) - \nabla_{xx} m_k^f(x_{k_i}) = \sum_{j=1}^n c_j(x_{k_i}) \nabla_{xx} c_j(x_{k_i})$$

from AC.3e, AM.4e, and the convergence of  $c_j(x_{k_i})$  to zero as  $i$  tends to infinity. AM.5 then follows immediately.  $\square$

This result shows that the Gauss–Newton model is asymptotically adequate (in the sense that it satisfies AM.5) *when convergence occurs to a root*.

The second observation is that Theorem 6.5.5 (p. 146) ensures that the trust-region constraint is asymptotically inactive, which has the following interesting consequence.

**Theorem 16.1.4** Suppose that AC.1e, AC.3e, AN.1, AM.1e–AM.4e, and AA.1 hold. Suppose also that there is a limit point  $x_*$  of the sequence of iterates generated by Algorithm BTR applied to problem (16.1.2) such that  $c(x_*) = 0$ . Suppose furthermore that the Gauss–Newton model (16.1.6) is used or that AM.5e holds. Suppose finally that  $A(x_*)$  has full rank. Then the complete sequence of iterates  $\{x_k\}$  converges to  $x_*$ , all iterations are eventually successful, and the trust-region radius  $\Delta_k$  is bounded away from zero. Moreover, if  $m = n$ , the trust region  $\mathcal{B}_k$  contains a root  $x_k^r$  of the model of  $c$  (that is,  $m_k^c(x_k^r) = 0$ ) for all  $k$  sufficiently large.

**Proof.** We first observe that our full-rank assumption on the Jacobian matrix implies that the Hessian of the objective function (16.1.4) is positive definite at  $x_*$ . Indeed, this immediately results from (16.1.5), the fact that, by assumption,  $c_i(x_*) = 0$  for all  $i$ , and the assumption that the rank of  $A(x_*)$  is full. We also note that Theorems 16.1.1 and 16.1.2 together guarantee that AF.1–AF.3 and AM.1–AM.4 hold. Moreover, Theorem 16.1.3 or the second conclusion of Theorem 16.1.2 ensures that AM.5 also holds. As a consequence, and since AA.1 holds by assumption, we may apply Theorem 6.5.5 (p. 146) and deduce that the complete sequence of iterates converges to  $x_*$ , that all iterations are eventually successful, and that the trust-region radius is bounded away from zero, that is, that

$$\Delta_k \geq \epsilon \tag{16.1.10}$$

for some  $\epsilon > 0$ .

Suppose now that  $m = n$  and consider the sequence of points  $x_k(j)$  recursively defined, for  $j \geq 0$ , by

$$x_k(0) = x_k, \quad s_k(j) = -A_k^{-1}m_k^c(x_k(j)), \quad \text{and} \quad x_k(j+1) = x_k(j) + s_k(j). \tag{16.1.11}$$

We first notice that this sequence is well defined for  $k$  sufficiently large, because the convergence of  $\{x_k\}$  to  $x_*$ , the continuity of the Jacobian, and our assumption

that  $A(x_*)$  has full rank together imply that  $A(x_k) = A_k$  also has full rank for  $k$  sufficiently large, where this last equality holds because of AM.3e.

Let us choose a  $\delta > 0$  small enough to ensure that

$$4\delta \|A(x_*)^{-1}\|_2 \leq \frac{\epsilon}{\kappa_{\text{une}}} \quad \text{and} \quad \delta \sqrt{n} \kappa_{\text{umh}} \|A(x_*)^{-1}\|_2^2 \leq \frac{1}{2}. \quad (16.1.12)$$

Suppose now that  $k$  is sufficiently large to ensure that

$$\|A_k^{-1}\|_2 \leq 2\|A(x_*)^{-1}\|_2, \quad (16.1.13)$$

which is possible because of AM.3e, the continuity of the Jacobian, and the convergence of  $\{x_k\}$  to  $x_*$ . Furthermore assume that

$$\|m_k^c(x_k(0))\|_2 = \|c(x_k)\|_2 \leq \frac{1}{2}\delta, \quad (16.1.14)$$

which is possible because of AM.2e and the fact that  $x_*$  is a root of  $c$ . Consider now a proof by induction. Assume that at iteration  $j$  of (16.1.11)

$$\|m_k^c(x_k(j))\|_2 \leq (\frac{1}{2})^{2^j} \delta \quad \text{and} \quad \|s_k(p)\|_2 \leq (\frac{1}{2})^{2^p-1} \delta \|A(x_*)^{-1}\|_2 \quad (p = 0, \dots, j-1). \quad (16.1.15)$$

The first of these conditions is clearly true for  $j = 0$  because of (16.1.14) and the second is empty for this value of  $j$ . Then, using the definition of  $s_k(j)$  in (16.1.11), (16.1.13), and (16.1.15), we deduce that

$$\begin{aligned} \|s_k(j)\|_2 &\leq \|A_k^{-1}\|_2 \|m_k^c(x_k(j))\|_2 \\ &\leq 2\|A(x_*)^{-1}\|_2 \|m_k^c(x_k(j))\|_2 \\ &\leq (\frac{1}{2})^{2^j-1} \delta \|A(x_*)^{-1}\|_2 \end{aligned} \quad (16.1.16)$$

for  $k$  large enough. As a consequence, we have that

$$\begin{aligned} \|x_k(j+1) - x_k\|_2 &\leq \sum_{p=0}^j \|s_k(p)\|_2 \\ &\leq \delta \|A(x_*)^{-1}\|_2 \sum_{p=0}^j (\frac{1}{2})^{2^p-1} \\ &\leq 4\delta \|A(x_*)^{-1}\|_2 \\ &\leq \frac{\epsilon}{\kappa_{\text{une}}}, \end{aligned}$$

where we have used the first part of (16.1.12). Because of (16.1.10) and AN.1, this yields that

$$\|x_k(j+1) - x_k\|_k \leq \Delta_k. \quad (16.1.17)$$

Thus  $x_k(j+1) \in \mathcal{B}_k$ . Furthermore, Taylor's theorem implies that

$$m_k^c(x_k(j+1)) = m_k^c(x_k(j)) + A_k s_k(j) + \frac{1}{2} \begin{pmatrix} \langle s_k(j), \nabla_{xx} m_{1k}^c(\xi_{1k}) s_k(j) \rangle \\ \vdots \\ \langle s_k(j), \nabla_{xx} m_{nk}^c(\xi_{nk}) s_k(j) \rangle \end{pmatrix} \quad (16.1.18)$$

for some  $\{\xi_{ik}\}_{i=1}^n$  in the segment  $[x_k(j), x_k(j+1)] \subset \mathcal{B}_k$ . We may thus deduce from (16.1.18), the Cauchy–Schwarz inequality, the definition of  $s_k(j)$ , AM.4e, and

(16.1.16) that

$$\begin{aligned}
 \|m_k^c(x_k(j+1))\|_2 &\leq \|(I - A_k A_k^{-1})m_k^c(x_k(j))\|_2 \\
 &\quad + \frac{1}{2}\sqrt{n} \max_{i=1,\dots,n} |\langle s_k(j), \nabla_{xx} m_{ik}^c(\xi_{ik}) s_k(j) \rangle| \\
 &\leq \frac{1}{2}\sqrt{n} \kappa_{\text{umh}} \|s_k(j)\|_2^2 \\
 &\leq \left(\frac{1}{2}\right)^{2^{j+1}-1} \delta^2 \sqrt{n} \kappa_{\text{umh}} \|A(x_*)^{-1}\|_2^2 \\
 &\leq \left(\frac{1}{2}\right)^{2^{j+1}} \delta,
 \end{aligned}$$

where we have used the second part of (16.1.12) to deduce the last inequality. We see from this bound and (16.1.16) that conditions (16.1.15) again hold for  $j+1$  and  $k$  sufficiently large, and thus for all  $j \geq 0$ . But (16.1.17) and the compactness of  $\mathcal{B}_k$  imply that the sequence  $\{x_k(j)\}$  has at least a limit point  $x_k^r \in \mathcal{B}_k$  for  $k$  sufficiently large, and the first part of (16.1.15) yields that  $m_k^c(x_k^r) = 0$ , which concludes the proof of the theorem.  $\square$

The point of Theorem 16.1.4 is not only that a minimizer of the residual norm asymptotically exists within the trust region, which is implied by the inactive nature of the trust-region constraint for large  $k$ , but also that this minimizer is actually a zero. Figure 16.1.2 shows the linearized equations and the associated contour lines in the trust region for the system of equations of Figure 16.1.1.

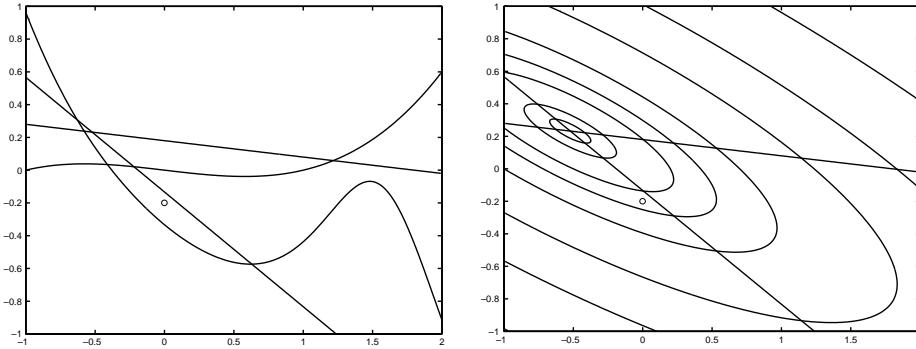


Figure 16.1.2: The linearized system of equations within the trust region (corresponding to the example of Figure 16.1.1) and the contour lines of the associated squared residual norm of the Gauss–Newton model (the current iterate  $x_k$  is denoted by a small circle).

Observe that  $x_k(1)$  in the proof above corresponds to a zero of the Gauss–Newton model and that  $x_k^r$  is obtained for a more general model by using a Gauss–Newton iteration while remaining in the trust region. This also suggests a possible algorithm for computing a good step when using a nonlinear model.

We conclude this section by noting that we could measure the residual of the nonlinear equation with measures other than the  $\ell_2$  norm. Our theory is still valid provided the resulting objective function remains twice-continuously differentiable. Other cases are discussed in the next sections.

## Notes and References for Section 16.1

Because the original papers of Levenberg (1944), Morrison (1960), and Marquardt (1963) were motivated by the solution of least-squares fitting problems, it is not surprising that many trust-region methods have evolved from these early proposals for this particular, but very important, area of optimization. The contributions of Fletcher (1971b), Reid (1973), Osborne (1976), Moré (1978), and Dennis, Gay, and Welsch (1981) were influential in making trust-region methods a central tool in the solution of least-squares problems, the first two resulting in **NS02**, **VA07**, and **VB01** within the Harwell Subroutine Library, while the last two led to MINPACK (see Moré, Garbow, and Hillstrom, 1980) and NL2SOL. All of these are high-quality packages for the solution of nonlinear least-squares problems.

An innovative element of Dennis, Gay, and Welsch's proposal is a mechanism that selects, at each iteration, either the Gauss–Newton or the pure Newton model. This technique is also used by Bunch, Gay, and Welsch (1993) in a package for the solution of general nonlinear statistical parameter estimation problems, such as maximum likelihood, maximum quasi-likelihood, generalized nonlinear least-squares, and some robust fitting problems. It also appears in the algorithm discussed by Toint (1987a, 1987b), where the trust-region mechanism is used in connection with partitioned quasi-Newton updating techniques for large-scale partially separable problems. Another feature of Toint's proposal, which is also embodied in his Harwell Subroutine Library, package **VE10**, is that the trust-region subproblem is only solved approximately, by a truncated conjugate gradient method. Similar ideas were considered by Martínez (1997), while Lukšan (1993) investigated using the LSQR linear least-squares algorithm of Paige and Saunders (1982) to generate approximate solutions to the trust-region subproblem. Conjugate-gradients-squared (CGS) were considered by Lukšan (1994) (see also Lukšan and Vlček, 1997), and a truncated conjugate gradient double-dogleg Lanczos process was proposed by Lukšan (1996a).

A number of other algorithmic improvements have also been considered in the context of least-squares problems, including the use of continuation techniques (see Salane, 1987), tensor models (see Hanson and Krogh, 1992, and Bouaricha and Schnabel, 1998, the latter resulting in the software package TENSOLVE of Bouaricha and Schnabel, 1997, 1999), special scaling (see Schwetlick and Tiller, 1989), decomposition techniques (see Lukšan, 1996b), hybrid method for large-residual or ill-conditioned problems (see Lukšan, 1996c), and the use of parallel computers (see Coleman and Plassman, 1992) and extensions to cover the case where  $c(x)$  may be nonsmooth (see de Sampaio, Yuan, and Sun, 1999).

Rank deficient problems often arise for a variety of “inverse” problems, where over-determined models are fitted to data. Techniques to handle such problems involve “regularization”: they are discussed, for example, by Vogel (1990), Rudnicki (1994), Hanke (1997), Martínez and Santos (1997), and Rojas (1998) and, in more general contexts, by Smith and Bowers (1993), Schaepperle and Luder (1994), and Petzold, Ren, and Maly (1997). The papers of Wright and Holt (1985), Bierlaire, Toint, and Tuyttens (1991), Sagara and Fukushima (1991), Hanson and Krogh (1992), and Sagara and Fukushima (1995) all deal with least-squares problems involving constraints on the parameters.

An especially interesting extension of the least-squares fitting problem is the *orthogonal distance regression* technique considered by Boggs, Byrd, and Schnabel (1987) (and also by Schwetlick and Tiller, 1989), where it is assumed that the input variables  $z_i$  (in our description on p. 749) are only correct to some unknown precision  $\epsilon_i$ , in which case the exact value of these errors has to be determined along with the vector  $x$ . The idea is again similar to that

developed above for the simple nonlinear fitting problem and results in a minimization problem of the form

$$\min_{x, \epsilon_i} \frac{1}{2} \sum_{i=1}^m w_i [h(x, z_i + \epsilon_i) - y_i]^2 + \langle \epsilon_i, S_i \epsilon_i \rangle$$

for some weights  $w_i > 0$  and some positive definite diagonal scaling matrices  $S_i$ , rather than (16.1.3). The ORDPACK package described by Boggs, Byrd, and Schnabel (1987) is an implementation of a trust-region method for this problem and has been applied to a variety of practical problems (see Mallick, 1997, for an example). The orthogonal distance regression method can be considered as a nonlinear version of the total least-squares approach, which is well established for linear data-fitting problems (see, for instance, Van Huffel and Vandewalle, 1991, or Van Huffel, 1997). See also Helfrich and Zwick (1996). Finally, we mention that the Levenberg–Morrison–Marquardt algorithm can be generalized to handle yet other kinds of parameter estimation problems, as illustrated by Osborne (1987) and Edlund, Ekblom, and Madsen (1997).

Methods for solving systems of nonlinear equations have always been intimately related to least-squares approaches, because the use of the  $\ell_2$  norm of the residual of the system provides a natural and easy measure of progress towards a root. The development of trust-region methods in this specific context started with Powell (1970c), which, as we mentioned on p. 9, sparked much of the later interest in such methods. This seminal contribution was followed by that of Reid (1973), based again on the ideas embodied in the Levenberg–Morrison–Marquardt algorithm. Among the more recent contributions, we may cite El-Hallabi (1987, 1990, 1993) and El-Hallabi and Tapia (1993), where the choice of the norm defining the trust region is relaxed, allowing, for instance, the use of general polyhedral norms (see the next section). This therefore encompasses the methods discussed above, but also some of the material contained in the rest of this chapter. Special attention is given to the large-scale case by Jarausch and Mackens (1987) and Lukšan (1994), while Lukšan and Vlček (1997) study the impact of using approximate Jacobian information, in the spirit of Section 8.4 (p. 280). The use of more complex tensor models is discussed by Bouaricha and Schnabel (1997), who show that such models may be very useful, especially when the Jacobian matrix happens to be singular at the root.

To conclude this brief literature overview, we should mention the papers by El-Hallabi and Tapia (1995) and Dennis, El-Alem, and Williamson (1999), who consider the solution of a system of mixed (possibly nonlinear) equalities and inequalities in the least-squares sense. One interesting aspect of the second of these is a trust-region method in which a model is chosen at each iteration from among a variety of possible models, depending on which of the linearized (in)equalities are active or violated at the Cauchy point. If the problem is reformulated by transforming inequalities into equalities using slack variables, it becomes clear that this technique is closest in spirit to the methods studied in Chapter 12, where the Cauchy point determines which of the bounds have to remain active at the trial points.

## 16.2 Nonlinear Equations in Other Norms

As we have seen, measuring the deviation of  $x$  from a root of  $c(x) = 0$  using  $\|c(x)\|_2$  has a number of advantages, mostly relating to the differentiability of the equivalent measure  $\|c(x)\|_2^2$ . In many applications, though, we are unlikely to be able to find a

root and will probably be satisfied instead by finding an  $x$  that makes  $\|c(x)\|_2$  as small as possible. If this is the case, there is nothing sacred about the  $\ell_2$  norm, and in many cases this is not the most appropriate norm to use. In this section, we consider some alternatives.

For the nonlinear fitting problem discussed at the start of Section 16.1, unfortunately, it often happens that a small number of the outputs are actually “blunders” or outliers—for instance, readings may have been recorded in error, an experiment may have been contaminated, or a machine may have inadvertently been reset—and ideally these rogue data points would be removed. The difficulty is that the residuals  $h(x, z_i) - y_i$  will inevitably be large for blunders, and these few residuals will tend to swamp the remaining accurate data leading to essentially worthless fits. In this case, it is preferable to give less weight to large residuals, and fitting in the  $\ell_1$  rather than the  $\ell_2$  norm is preferable. At the other extreme, there may be good reasons why the overriding consideration is that the largest residual, and therefore all the residuals, should be as small as possible. If this is the goal, the  $\ell_\infty$  norm should be used. Although there are occasionally cases where an  $\ell_p$  norm ( $p \neq 1, 2, \infty$ ) is appropriate,<sup>271</sup> by far the majority of fitting takes place in these three norms. Figure 16.2.1 shows the contour lines of the residual measured in the  $\ell_1$  and  $\ell_\infty$  norms for the problem of Figure 16.1.1.

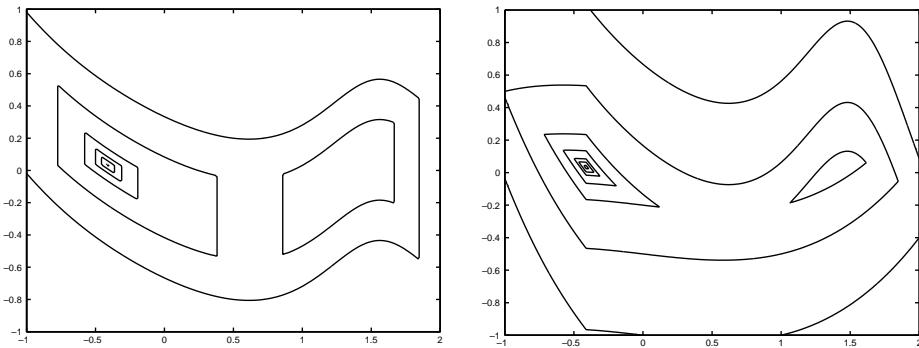


Figure 16.2.1: The contour lines of the  $\ell_1$  (left) and  $\ell_\infty$  (right) norms of the residual for the problem of Figure 16.1.1.

As we have already considered the  $\ell_2$  norm in Section 16.1, we shall concentrate here on the minimization of convex, polyhedral functions

$$h(c) = \max_{1 \leq i \leq \ell} \ell_i(c), \quad (16.2.1)$$

for which each contribution

$$\ell_i(c) = \langle p_i, c \rangle + b_i \quad (16.2.2)$$

is linear. Clearly the  $\ell_1$  and  $\ell_\infty$  norms are prime examples (see Section 11.4), but this generalization also allows us to cover the related and equally important class of

---

<sup>271</sup>See, for example, Brimberg and Love (1991).

*min-max*<sup>272</sup> problems for which

$$h(c) = \max_{1 \leq i \leq m} c_i(x).$$

### 16.2.1 Global Convergence

Much of the groundwork in these cases has been covered in Chapter 11 and Section 15.3.2. We aim to minimize the nonsmooth function

$$f(x) = h(c(x)), \quad (16.2.3)$$

where  $h(c)$  is given by (16.2.1). We may do so using Algorithm 11.1.1 (p. 413) (or its nonmonotone counterpart) or Algorithm 11.3.1 (p. 425), with the model

$$m(x, H, s) = \frac{1}{2} \langle s, Hs \rangle + \max_{1 \leq i \leq \ell} \ell_i(c(x) + A(x)s), \quad (16.2.4)$$

where, as before,  $A(x) \stackrel{\text{def}}{=} \nabla_x c(x)$ ,  $H$  is an approximation of  $\sum_{i=1}^m y_i \nabla_{xx} c_i(x)$ , and  $y$  lies close to the generalized gradient  $\partial h(c(x))$  (see Section 11.5). We then have our central global convergence result.

**Theorem 16.2.1** Suppose that AC.1e holds and that the sequence  $\{x_k\}$  is generated by Algorithm 11.1.1 (or its nonmonotone counterpart or Algorithm 11.3.1) applied to the minimization of (16.2.3). Suppose further that we use the model (16.2.4) whose Hessian  $H_k$  satisfies AM.4j, with steps chosen to satisfy AA.1n, and that  $\{x_k\}$  has a limit point  $x_*$ . Then  $x_*$  is a first-order critical point for (16.2.3).

**Proof.** Assumption AM.4j is sufficient for Theorem 11.5.1 (p. 436) to ensure that the model (16.2.4) satisfies the assumptions of Theorems 11.2.5 (p. 420) and 11.3.2 (p. 426), and thus that  $x_*$  is a first-order critical point for (16.2.3).  $\square$

### 16.2.2 Asymptotic Convergence of the Basic Method

We have seen that the global convergence properties of Algorithm 11.1.1 and its variants are satisfactory. But what of the asymptotic convergence behaviour? Unfortunately, this is far from perfect, even if we solve the trust-region subproblem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad m(x, H, s) \quad \text{subject to} \quad \|s\|_2 \leq \Delta \quad (16.2.5)$$

exactly. Let us assume that, in addition to AM.4j,

---

<sup>272</sup>These are sometimes called minimax or Chebychev problems.

**AI.2e** the sequence  $\{x_k\}$  generated by Algorithm 11.1.1 (p. 413) (or its nonmonotone counterpart or Algorithm 11.3.1 (p. 425)) applied to the minimization of (16.2.3) with steps chosen to satisfy AA.1e, has a single limit point  $x_*$ ,

and

**AA.1e** the step  $s$  is chosen as a local minimizer of the model (16.2.5) for which AA.1n holds.

To say more, we shall require that suitable analogs of AO.1b, AO.3, and AO.4 hold at  $(x_*, y_*)$ . We first consider AO.3. One way of deriving a suitable condition is simply to consider the special case of AO.3n when  $f(x) = 0$ . This leads to us requiring that

**AO.3e** the relationships

$$A^T(x_*)y_* = 0 \quad (16.2.6)$$

and

$$\left\langle s, \sum_{i=1}^n [y_*]_i \nabla_{xx} c_i(x_*) s \right\rangle > 0 \quad (16.2.7)$$

hold for all  $s$  in

$$\mathcal{D} = \left\{ d \mid \max_{y \in \partial h(c(x_*))} \langle d, (\nabla_x c(x_*))^T y \rangle = 0 \text{ and } \|d\|_2 = 1 \right\} \quad (16.2.8)$$

at  $x_* \in \mathcal{X}$  and  $y_* \in \partial h(c(x_*))$ .

Another, rather revealing, way of deriving this condition is to restate the problem of minimizing (16.2.3) as the equivalent nonlinear programming reformulation

$$\underset{(x,h) \in \mathbb{R}^{n+1}}{\text{minimize}} \quad h \quad \text{subject to} \quad \ell_i(c(x)) \leq h \quad \text{for } 1 \leq i \leq \ell. \quad (16.2.9)$$

We then have the following result.

**Theorem 16.2.2** The second-order sufficiency conditions AO.3 for (16.2.9) are equivalent to the second-order sufficiency conditions AO.3e for the problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad h(c(x)) \quad (16.2.10)$$

when  $h(c)$  has the form (16.2.1).

**Proof.** The appropriate second-order sufficiency conditions (3.2.5)–(3.2.8) (pp. 40, 41) and (3.2.11) (p. 41) for (16.2.9) are that  $x_*$  and some Lagrange multipliers  $\theta_*$  satisfy the conditions

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \sum_{i=1}^{\ell} [\theta_*]_i \begin{pmatrix} -A^T(x_*) p_i \\ 1 \end{pmatrix}, \quad (16.2.11)$$

$$h_* - \ell_i(c(x_*)) \geq 0 \text{ for all } 1 \leq i \leq \ell, \quad (16.2.12)$$

$$[\theta_*]_i \geq 0 \text{ for all } 1 \leq i \leq \ell, \quad (16.2.13)$$

$$(h_* - \ell_i(c(x_*))) [\theta_*]_i = 0 \text{ for all } 1 \leq i \leq \ell, \quad (16.2.14)$$

$$\text{and } \left\langle s, \sum_{i=1}^{\ell} [P\theta_*]_i \nabla_{xx} c_i(x_*) s \right\rangle > 0 \text{ for all } \begin{pmatrix} s \\ s^h \end{pmatrix} (\neq 0) \in \mathcal{N}_+^h. \quad (16.2.15)$$

Here  $P$  is the matrix with columns  $p_i$ ,  $1 \leq i \leq \ell$ ,

$$\mathcal{N}_+^h = \left\{ \begin{pmatrix} s \\ s^h \end{pmatrix} \in \mathbb{R}^{n+1} \mid \begin{array}{l} -\langle s, A^T(x_*) p_i \rangle + s^h = 0 \text{ for all } i \in \mathcal{A}(x_*) \mid [\theta_*]_i > 0 \\ \text{and} \\ -\langle s, A^T(x_*) p_i \rangle + s^h \geq 0 \text{ for all } i \in \mathcal{A}(x_*) \mid [\theta_*]_i = 0 \end{array} \right\},$$

with the set of active contributions

$$\mathcal{A}(x_*) \equiv \mathcal{A}(c(x_*)) = \{i \mid \ell_i(c(x_*)) = h(c(x_*)) \text{ and } 1 \leq i \leq \ell\}.$$

On writing

$$y_* = P\theta_*, \quad (16.2.16)$$

we see from (16.2.11) that (16.2.6) holds, while the same equation and (16.2.13) give that

$$\sum_{i=1}^{\ell} [\theta_*]_i = 1 \text{ and } \theta_* \geq 0, \quad (16.2.17)$$

and from (16.2.12)–(16.2.14) that

$$[\theta_*]_i = 0 \text{ for all } i \notin \mathcal{A}(c(x_*)). \quad (16.2.18)$$

Thus it follows from Theorem 11.4.3 (p. 432) that  $y_* \in \partial h(c(x_*))$ , and hence from (16.2.6) and Corollary 3.2.13 (p. 48) that  $(x_*, y_*)$  satisfies the first-order necessary conditions for (16.2.10) if and only if  $(x_*, \theta_*)$  does the same for (16.2.9).

Now consider any  $s$  for which  $(s, s^h)$  lies in  $\mathcal{N}_+^h$ . If this is the case, we have that

$$\langle s, A^T(x_*) [\theta_*]_i p_i \rangle = [\theta_*]_i s^h \text{ for all } i \in \mathcal{A}(x_*),$$

and hence, from (16.2.16), (16.2.17), and (16.2.18), that

$$\begin{aligned} 0 &= \langle s, A^T(x_*) y_* \rangle = \left\langle s, A^T(x_*) \sum_{i=1}^{\ell} [\theta_*]_i p_i \right\rangle = \sum_{i=1}^{\ell} \langle s, A^T(x_*) [\theta_*]_i p_i \rangle \\ &= \sum_{i \in \mathcal{A}(x_*)} \langle s, A^T(x_*) [\theta_*]_i p_i \rangle = \sum_{i \in \mathcal{A}(x_*)} [\theta_*]_i \langle s, A^T(x_*) p_i \rangle = \sum_{i \in \mathcal{A}(x_*)} [\theta_*]_i s^h = \sum_{i=1}^{\ell} [\theta_*]_i s^h \\ &= s^h. \end{aligned}$$

Thus  $s^h = 0$ , and (16.2.15) is equivalent to requiring that  $s$  satisfies (16.2.7), where

$$\mathcal{N}_+ = \left\{ s \in \mathbb{R}^n \mid \begin{array}{l} \langle s, A^T(x_*) p_i \rangle = 0 \text{ for all } i \in \mathcal{A}(x_*) \mid [\theta_*]_i > 0 \text{ and} \\ \langle s, A^T(x_*) p_i \rangle \leq 0 \text{ for all } i \in \mathcal{A}(x_*) \mid [\theta_*]_i = 0 \end{array} \right\}.$$

If  $s \in \mathcal{N}_+$ , we have that

$$\langle s, A^T(x_*)p_i \rangle \leq 0 \text{ for all } i \in \mathcal{A}(x_*). \quad (16.2.19)$$

Therefore, if  $y \in \partial h(c(x_*))$ , we may write  $y = \sum_{i \in \mathcal{A}(x_*)} \theta_i p_i$ , and hence it follows that

$$\langle s, A^T(x_*)y \rangle \leq 0, \quad (16.2.20)$$

with equality when  $y = y_*$ . We then have that  $s \in \mathcal{D}$ , where  $\mathcal{D}$  is given by (16.2.8). Conversely, if  $s \in \mathcal{D}$ , (16.2.20) holds for all  $y \in \partial h(c(x_*))$ . Since  $p_i \in \partial h(c(x_*))$  when  $i \in \mathcal{A}(x_*)$ , we then have that (16.2.19) holds. For such an  $s$ , consider an index  $j \in \mathcal{A}(x_*)$  for which  $[\theta_*]_j > 0$ , and suppose that, for this  $j$ ,  $\langle s, A^T(x_*)p_j \rangle < 0$ . Then (16.2.6), (16.2.17), and (16.2.19) show that

$$0 = \langle s, A^T(x_*)y_* \rangle = \sum_{i=1}^{\ell} [\theta_*]_i \langle s, A^T(x_*)p_i \rangle \leq [\theta_*]_j \langle s, A^T(x_*)p_j \rangle < 0.$$

This contradiction shows that  $\langle s, A^T(x_*)p_i \rangle = 0$  whenever  $[\theta_*]_i > 0$ , and this combines with (16.2.19) to show that  $s \in \mathcal{N}_+$ . Hence  $(x_*, y_*)$  satisfies the second-order sufficiency conditions for (16.2.10) if and only if  $(x_*, \theta_*)$  does the same for (16.2.9).

□

It immediately follows from Theorems 3.2.14 (p. 49) and 16.2.2 that  $x_*$  is a strict, isolated minimizer of  $h(c(x))$  if AM.4n holds.

The appropriate versions of AO.1b and AO.4 are as follows.

**AO.1e** The vectors

$$\{A^T(x_*)(p_i - p_j)\}_{i \in \mathcal{A}(x_*) \setminus \{j\}}$$

are linearly independent, where  $j$  is any member of  $\mathcal{A}(x_*)$ ,

and the  $p_i$  are as in (16.2.2).

**AO.4e** The generalized gradient  $y_*$  lies in the interior of  $\partial h(c(x_*))$ .

Assumption AO.1e is equivalent to requiring that the vectors

$$\begin{pmatrix} -A^T(x_*)p_i \\ 1 \end{pmatrix}, \quad i \in \mathcal{A}(x_*),$$

are linearly independent, which is precisely the constraint qualification AO.1b appropriate for the nonlinear programming reformulation (16.2.9). Likewise, AO.4e is equivalent to requiring that  $[\theta_*]_i > 0$  for all  $i \in \mathcal{A}(x_*)$ , which is AO.4 for (16.2.9). Thus AO.1b, AO.3, and AO.4 all have natural equivalents for (16.2.10), and each can be interpreted as the relevant assumption on a local solution to (16.2.9). This is extremely useful since we can now analyse the local convergence of algorithms for (16.2.10) purely in terms of equivalent algorithms for (16.2.9).

We shall also assume, for the time being, that

**AA.11e** the sequence  $\{\Delta_k\}$  generated by Algorithm 11.1.1 (p. 413) (or its nonmonotone counterpart or Algorithm 11.3.1 (p. 425)) applied to the minimization of (16.2.10) with steps chosen to satisfy AA.1e, is such that

$$\Delta = \liminf_{k \rightarrow \infty} \Delta_k > 0.$$

It is then clear that if AA.11e holds, the trust-region constraint will be inactive for all sufficiently large  $k$ , and we may disregard the trust-region constraint in (16.2.5) without affecting the solution. In this case, Theorem 16.2.2 implies that the subproblem (16.2.5) is equivalent to the quadratic program

$$\underset{(s, s^h) \in \mathbb{R}^{n+1}}{\text{minimize}} \quad \frac{1}{2} \langle s, Hs \rangle + s^h \quad \text{subject to} \quad \ell_i(c(x) + A(x)s) \leq s^h \quad \text{for } 1 \leq i \leq \ell \quad (16.2.21)$$

under AM.4n and AA.11e. Just as we saw in Section 15.3.2, Theorem 3.2.7 (p. 42) shows that AO.1b, AO.3, and AO.4 hold for the solution  $(s, s^h)$  of (16.2.21) for all  $k$  sufficiently large. So long as

**AA.1f** the step  $s_k$  is ultimately chosen as the locally unique minimizer of the model (16.2.5) for all  $x_k - x_*$  sufficiently small,

Theorem 16.2.2 implies that  $(s_k, s_k^h)$  is also a strict, isolated local minimizer of (16.2.21) for all large  $k$ , and thus the iteration is equivalent to the SQP method analysed in Section 15.2.2 applied to the nonlinear programming reformulation (16.2.9). Hence, under the stated assumptions, in view of Theorem 15.2.2 (p. 632) we would then expect that the iteration will converge Q-superlinearly given suitable Lagrange multiplier estimates.<sup>273</sup> But, as in Section 15.3.2, it is assumption AA.11e that we must question.

For consider the problem

$$\begin{aligned} \underset{x_1, x_2}{\text{minimize}} \quad & \max \left[ 4(x_1^2 + x_2^2 - 1) - x_1, -x_1 \right] \\ \equiv \quad & 2(x_1^2 + x_2^2 - 1) - x_1 + \max \left[ 2(x_1^2 + x_2^2 - 1), 2(1 - x_1^2 - x_2^2) \right] \\ = \quad & 2(x_1^2 + x_2^2 - 1) - x_1 + 2|x_1^2 + x_2^2 - 1|. \end{aligned} \quad (16.2.22)$$

---

<sup>273</sup>Note that, strictly, we have shown that  $(x_k, h_k)$  converges Q-superlinearly to  $(x_*, h(c(x_*)))$  under the given assumptions. It is actually possible to show that  $x_k$  converges Q-superlinearly to  $x_*$  under the same assumptions. To do so, AO.1e, AO.3e, and AO.4e suffice to show that (16.2.9) is equivalent to

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \langle y_*, c(x) \rangle \quad \text{subject to} \quad \ell_i(c(x)) = \ell_j(c(x)) \quad \text{for all } i \in \mathcal{A}(x_*) \setminus \{j\},$$

and (16.2.21) is equivalent to

$$\begin{aligned} \underset{s \in \mathbb{R}^n}{\text{minimize}} \quad & \frac{1}{2} \langle s, Hs \rangle + \langle y_*, c(x) + A(x)s \rangle \\ \text{subject to} \quad & \ell_i(c(x) + A(x)s) = \ell_j(c(x) + A(x)s) \quad \text{for all } i \in \mathcal{A}(x_*) \setminus \{j\}, \end{aligned}$$

where  $j$  is any member of  $\mathcal{A}(x_*)$ , when  $x$  is sufficiently close to  $x_*$ ; both of the above follow by using (any) one of the active contributions to eliminate  $h$  and  $s^h$ , respectively. The above subproblem can be shown to satisfy the relevant versions of AO.1b, AO.3, and AO.4 and to have the same solutions as (16.2.5), and thus Theorem 15.2.2 (p. 632) implies Q-superlinear convergence of  $x_k$ .

This is the  $\ell_1$  exact penalty function for problem (15.3.10) (p. 641) considered in Section 15.3.2.2 with the penalty parameter  $\sigma = 2$ , which is larger than the critical Lagrange multiplier  $\frac{3}{2}$ , but it is also a polyhedral convex function of the form (16.2.1). As before, it is easy to show that AO.1e, AO.3e, and AO.4e hold at the solution  $x_* = (1, 0)^T$  of (16.2.22), that both contributions  $c_1(x) = 4(x_1^2 + x_2^2 - 1) - x_1$  and  $c_2(x) = -x_1$  are active, and that the optimal generalized gradient is  $y_* = (\frac{1}{8}, \frac{7}{8})^T$ . We would expect that picking the model Hessian  $H_k$  for the subproblem (16.2.5) as the optimal value  $\sum_{i=1}^2 [y_*]_i \nabla_{xx} c_i(x_*) = I$  will lead to superlinear convergence. Since the model problem (16.2.5) and the SQP subproblem (15.2.5) (p. 624) have the same solution for all  $x_k = (\cos \theta, \sin \theta)^T$  sufficiently close to  $x_*$ , we have, as before, that  $s_k = (\sin^2 \theta, -\cos \theta \sin \theta)^T$  solves (16.2.5) so long as AA.11e holds. But unfortunately

$$h(c(x_k + s_k)) - h(c(x_k)) = 3 \sin^2 \theta > 0,$$

and thus the step  $x_k + s_k$  will not be accepted by any algorithm that requires monotonic descent at every successful iteration regardless of how close  $x_k$  is to  $x_*$ . Thus the Maratos effect strikes again (see Figure 16.2.2), and we must think harder if we are to develop a globally convergent method that ultimately converges at a satisfactory rate.

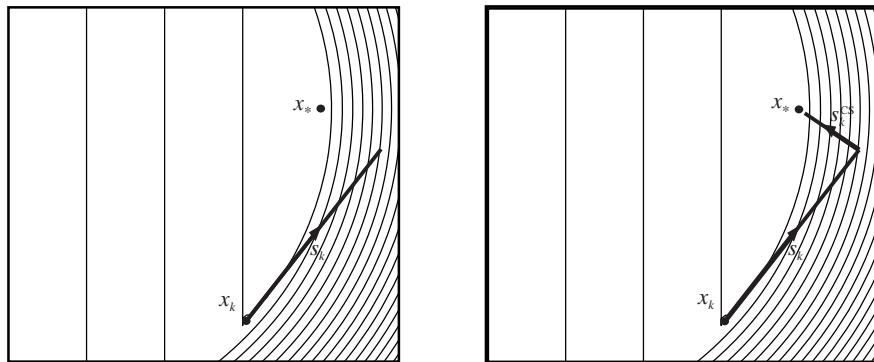


Figure 16.2.2: The figure on the left illustrates the Maratos effect for the problem (16.2.22). Notice how the objective function increases at the point  $x_k + s_k$ . See how the second-order correction in the figure on the right counteracts the Maratos effect.

### 16.2.3 Second-Order Corrections

The key is, once again, to correct the search direction for second-order changes absent from a first-order model of the contributions  $l_i(c(x))$ . The secret is to notice that not only are the problems (16.2.10) and (16.2.9) equivalent under AO.3e, but additionally  $h(c(x))$  may be regarded as an exact penalty function for (16.2.9).

To see this, consider the  $\ell_\infty$  exact penalty function

$$\phi((x, h), \sigma) = h + \sigma \max \left[ 0, \max_{1 \leq i \leq m} (\ell_i(c(x)) - h) \right]$$

$$\begin{aligned}
&= (1 - \sigma)h + \sigma \max \left[ h, \max_{1 \leq i \leq m} \ell_i(c(x)) \right] \\
&= (1 - \sigma)h + \sigma \max[h, h(c(x))]
\end{aligned} \tag{16.2.23}$$

for (16.2.9), where Theorem 14.5.1 (p. 610) requires that  $\sigma > \|\theta_*\|_1$ , which, in view of (16.2.17), is simply that  $\sigma > 1$ .

We then have the following result.

**Lemma 16.2.3** Suppose that  $\sigma > 1$ . Then, for any  $x$ ,

$$h(c(x)) = \min_h \phi((x, h), \sigma) = \arg \min_h \phi((x, h), \sigma). \tag{16.2.24}$$

**Proof.** Suppose first that  $h(c(x)) \leq h$ . In this case, (16.2.23) gives that

$$\phi((x, h), \sigma) = (1 - \sigma)h + \sigma h = h.$$

Thus

$$\min_h \phi((x, h), \sigma) = \min_{h \geq h(c(x))} h = h(c(x)),$$

with the minimizer being  $h = h(c(x))$ . Conversely, suppose that  $h(c(x)) \geq h$ . It then follows from (16.2.23) that

$$\phi((x, h), \sigma) = (1 - \sigma)h + \sigma h(c(x)).$$

In this case, writing  $w = -h$ ,

$$\begin{aligned}
\min_h \phi((x, h), \sigma) &= \sigma h(c(x)) + (\sigma - 1) \min_{-h(c(x)) \leq -h} (-h) \\
&= \sigma h(c(x)) + (\sigma - 1) \min_{-h(c(x)) \leq w} (w) \\
&= \sigma h(c(x)) + (\sigma - 1)(-h(c(x))) \\
&= h(c(x)),
\end{aligned}$$

where the minimizer is  $-h = w = -h(c(x))$ . Thus (16.2.24) holds in all cases.  $\square$

This then suggests an obvious approach.

At any iterate  $x_k$ , we may associate an  $h_k = h(c(x_k))$ . Suppose we attempt to solve the nonlinear programming reformulation (16.2.9) using the exact penalty function (16.2.23). Then we have seen in Section 15.3.2.3 that we may ensure an asymptotic reduction in (16.2.23) by adding a second-order correction to the appropriate SQP step. In particular, Theorem 15.3.7 (p. 651) shows that, for any  $\eta_2 < 1$ ,

$$\frac{\phi((x, h), \sigma) - \phi((x + s + s^{cs}, h + s^h + s^{hc}), \sigma)}{m_h((x, h), H, 0) - m_h((x, h), H, (s, s^h))} > \eta_2 \tag{16.2.25}$$

for all  $(x, y)$  that are sufficiently close to  $(x_*, y_*)$  provided that the SQP step  $(s, s^h)$  and the second-order correction  $(s^{cs}, s^{hc})$  are chosen appropriately. The model  $m_h$  here is simply

$$\begin{aligned} m_h((x, h), H, (s, s^h)) &= (1 - \sigma)(h + s^h) + \frac{1}{2}\langle s, Hs \rangle \\ &\quad + \sigma \max \left[ h + s^h, \max_{1 \leq i \leq m} \ell_i(c(x) + A(x)s) \right], \end{aligned}$$

and since the SQP step is intended to (locally) minimize this model, it is again straightforward to show that

$$\begin{aligned} m_h((x, h), H, (s, s^h)) &= \frac{1}{2}\langle s, Hs \rangle + \max_{1 \leq i \leq m} \ell_i(c(x) + A(x)s) \\ &= m(x, H, s). \end{aligned} \tag{16.2.26}$$

Furthermore, since we have chosen  $h = h(c(x))$ , we immediately have that

$$m_h((x, h), H, 0) = m(x, H, 0) \quad \text{and} \quad \phi((x, h), \sigma) = h(c(x)),$$

while (16.2.24) shows that

$$h(c(x + s + s^{cs})) \leq \phi((x + s + s^{cs}, h + s^h + s^{hc}), \sigma). \tag{16.2.27}$$

Hence, substituting (16.2.26) and (16.2.27) into (16.2.25), we find that

$$\frac{h(c(x)) - h(c(x + s + s^{cs}))}{m(x, H, 0) - m(x, H, s)} \geq \eta_2 \tag{16.2.28}$$

for all  $(x, y)$  that are sufficiently close to  $(x_*, y_*)$  provided that the SQP and second-order correction steps for the nonlinear programming reformulation (16.2.9) are used. But the components  $s$  and  $s^{cs}$  from these calculations may be equivalently obtained by minimizing the models (16.2.4) and

$$\frac{1}{2}\langle s^{cs}, Hs^{cs} \rangle + \max_{1 \leq i \leq \ell} \ell_i(c(x + s) + A(x + p)s^{cs}),$$

respectively, where  $p$  is any vector satisfying (15.3.39) (p. 649); globally, of course, the model problems will also include appropriate trust regions. It is important that the artificial variable  $h + s^h + s^{hc}$  be discarded after each step. Indeed, one may view the preferred value  $h = h(x + s + s^{cs})$  as a magical step (see Section 10.4.1) since the artificial merit function (16.2.23) may be improved at no computational cost with this value. It is also important to note that the solution  $s$  to our subproblem does not depend on the value of  $h$ .

The remaining details are now essentially the same as those outlined in Section 15.3.2.3. The second-order correction is sufficiently small that it will not interfere with the final Q-superlinear convergence rate of the basic step. The estimate (16.2.28) implies that the second-order correction provides a reduction in  $h(c(x))$ , so long as the trust-region radius does not interfere, for any  $(x, y)$  sufficiently close to  $(x_*, y_*)$ , and thus all that is necessary is to ensure that the trust-region radius is sufficiently large at one such point. This may be achieved by embedding the basic and second-order correction steps within a framework like Algorithm 11.3.1 (p. 425), with the actual step chosen using the following scheme.

**Algorithm 16.2.1: Possible Step 2 for Algorithm 11.3.1 (p. 425)**

**Step 2: Compute basic step.** Compute a step  $s_k$  satisfying AA.1n that reduces the model

$$m(x_k, H_k, s) = \frac{1}{2} \langle s, H_k s \rangle + \max_{1 \leq i \leq \ell} \ell_i(c(x_k) + A(x_k)s)$$

and for which  $x_k + s_k \in \mathcal{B}_k$ .

**Step 2a: Compute a correction.** If desired, compute a correction step  $s_k^{\text{CS}}$  that reduces the correction model

$$\begin{aligned} m^C(x_k, (H_k^C, s_k, g_k, t_k), s^{\text{CS}}) &= \frac{1}{2} \langle s^{\text{CS}}, H_k^C s^{\text{CS}} \rangle \\ &\quad + \max_{1 \leq i \leq \ell} \ell_i(c(x_k + s_k) + A(x_k + p_k)s^{\text{CS}}) \end{aligned}$$

for appropriate vectors  $g_k$  and  $p_k$  and matrix  $H_k^C$ , and for which

$$\|s_k + s_k^{\text{CS}}\|_k \leq \tau \Delta_k.$$

If

$$h(c(x_k + s_k + s_k^{\text{CS}})) > h(c(x_k + s_k)),$$

reset  $s_k^{\text{CS}} = 0$ .

**Step 2b: Skip the correction.** Otherwise, set  $s_k^{\text{CS}} = 0$ .

As before, there are good reasons to prefer Step 2b to Step 2a, at least when it is clear that the current iterate is far from the solution or when the basic step predicts an acceptable decrease in  $h(c(x))$ . The easiest way of ensuring that the radius will be inactive at some stage near the solution is, once again, to require that all successful steps are very successful, by setting  $\eta_1 = \eta_2$ , and to ensure that the updated radius following a very successful step is bounded away from zero, by setting  $\Delta_{k+1}$  according to (15.3.48) (p. 653). However, such rules may be rather restrictive, and it is better simply to ensure that they are adopted if there is any suspicion that the asymptotics are approaching. Although more subtle variations are possible, their analyses are considerably more complicated, and we will not give details here.

## Notes and References for Section 16.2

Li (1994b, 1994a) studied a trust-region algorithm for the solution of the nonlinear  $\ell_1$  problem that uses affine-scaling techniques inspired by those described in Section 13.12. One of the interests in using the  $\ell_1$  norm is it provides, in statistical applications, an indirect technique for variable selection in possibly overspecified models. This is advocated by Osborne (1998), where the contrast is also shown, from this point of view, with the  $\ell_2$  and  $\ell_\infty$  norms. The idea of variable selection using the  $\ell_1$  norm is also discussed, in a broader context, by Tibshirani

(1996). Another trust-region algorithm for nonlinear  $\ell_1$  problems is discussed by Gao (1998), who uses a sequence of problems involving the Huber robust regression function to approximate the absolute value. Yuan (1998d) considers using the  $\ell_\infty$  norm instead.

For historical reasons, most of the work on min-max problems is linesearch rather than trust region based. An early basic reference is Dem'yanov and Malozemov (1974). A recent detailed text is Polak (1997). There are more texts that address linear approximations in both the  $\ell_1$  and  $\ell_\infty$  norms; see, for example, Powell (1981a), Osborne (1985), and Watson (1980). Engineering applications of min-max include antenna and amplifier design and minimizing noise and static tuning in circuits (see, for example, Conn et al., 1996, Bakr et al., 1998, and Conn and Charalambous, 1975). Although the restatement of the problem as in (16.2.9) might suggest that there is not much innovation required to tackle such problems, many of them arise from discretizations of the continuous min-max problem

$$\min_x \max_{t \in T} \|\phi(x, t)\|,$$

where  $T$  is a compact set. Classical Chebychev theory provides characterizations for best linear approximations. They and generalizations of them need to be exploited for efficient algorithms; see, for example, Conn and Li (1992).

Nocedal (1984) and Duff, Nocedal, and Reid (1987) consider solving sparse systems of nonlinear equations via the  $\ell_1$  norm using an  $\ell_\infty$  trust region. Yuan (1996) shows that the latter method above can converge to a nonoptimal solution and supplies a suitable modification to ensure global convergence. Vardi (1992) uses the reformulation (16.2.9). Yuan (1984) gives an example of a trust-region approach to the min-max problem for which the Maratos effect persists and for which only a linear rate of convergence occurs.

Zhou and Tits (1993) describe a linesearch-based nonmonotone method for the min-max problem. Charalambous (1979) solves the problem by approximating it by sequences of least  $p$ th problems for increasing values of  $p$ , again using a linesearch method.

## 16.3 Complementarity Problems

### 16.3.1 The Problem and an Associated Merit Function

Having studied the solution of systems of nonlinear equations, we now turn to a different, but related, problem, called the *complementarity problem*.

Given two smooth functions  $g$  from  $\mathbb{R}^n$  into  $\mathbb{R}^n$  and  $h$  from  $\mathbb{R}^n$  into  $\mathbb{R}^n$ , the aim is to find a vector  $x \in \mathbb{R}^n$  for which

$$\langle g(x), h(x) \rangle = 0, \quad g(x) \geq 0, \quad \text{and} \quad h(x) \geq 0, \quad (16.3.1)$$

the last two inequalities being understood componentwise. The form (16.3.1) is known as the *generalized complementarity problem*, while the particular case where  $h(x) = x$ , that is,

$$\langle g(x), x \rangle = 0, \quad g(x) \geq 0, \quad \text{and} \quad x \geq 0,$$

is usually referred to as the *nonlinear complementarity problem* (NCP). Finally, if  $g$  is a linear map, that is, if

$$g(x) = a + Mx$$

for some vector  $a \in \mathbb{R}^n$  and some  $n \times n$  matrix  $M$ , we then have the *linear complementarity problem* (LCP). Further classes of LCP can be defined by making specific assumptions on  $M$ , but we will not need this refinement in the discussion that follows. To motivate the study of the complementarity problem, we note that the NCP contains the expressions for first-order criticality for a nonlinear optimization problem subject to nonnegativity constraints. In this case,  $g(x)$  is merely the gradient<sup>274</sup> of the objective function at  $x$ . When the problem is strictly convex, solving the NCP or the original problem is thus equivalent, because the only first-order critical point must then be the unique solution to the problem.<sup>275</sup>

Various methods have been proposed for solving problem (16.3.1), or the NCP. In our presentation, we naturally focus on techniques based on the trust-region idea. These techniques are based on the use of a *complementarity merit function*<sup>276</sup>  $\phi(a, b)$  from  $\mathbb{R}^2$  into  $\mathbb{R}$ , which is characterized by the following property of its roots: we must have that

$$\phi(a, b) = 0 \text{ if and only if } a \geq 0, \quad b \geq 0, \quad \text{and } ab = 0.$$

Such functions include the Fischer–Burmeister function

$$\phi(a, b) = \sqrt{a^2 + b^2} - a - b \tag{16.3.2}$$

or the residual function

$$\phi(a, b) = \min[a, b],$$

among many other possible choices. The level curves of these two functions are shown in Figure 16.3.1. In what follows, our analysis focuses on the use of the Fischer–Burmeister function (16.3.2) because, as we will see shortly, it has advantageous differentiability properties. Using this function, we may then rewrite the generalized complementarity problem in the form

$$\Phi(x) = 0, \tag{16.3.3}$$

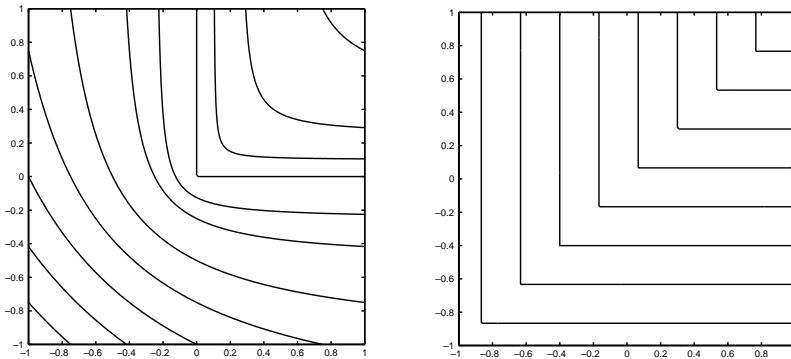


Figure 16.3.1: The Fischer–Burmeister (left) and the natural residual (right) complementarity merit functions.

<sup>274</sup>Notice that the objective function itself does not feature in this framework.

<sup>275</sup>See Section 3.2.3.

<sup>276</sup>Also called an *NCP function*.

where we have defined  $h_i(x) = [h(x)]_i$  and  $g_i(x) = [g(x)]_i$  for  $i = 1, \dots, n$ , and

$$[\Phi(x)]_i \stackrel{\text{def}}{=} \phi(h_i(x), g_i(x)).$$

We now see the connection with the rest of this chapter, in that (16.3.3) is nothing but a (nonsmooth) system of nonlinear equations. We may then think of applying the trust-region techniques for such problems, taking care of the nonsmoothness of  $\Phi$ . The function  $\Phi(x)$  is locally Lipschitz continuous on  $\mathbb{R}^n$  and differentiable on the set

$$\mathcal{D} \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid h_i^2(x) + g_i^2(x) > 0 \text{ for } i = 1, \dots, n\},$$

but not necessarily on the boundary of this set. In fact, it can be shown that the generalized Jacobian of  $\Phi$  at  $x$  is contained in the set of  $n \times n$  matrices  $V(x)$  given by

$$V(x) = \nabla_x h(x) D^h + \nabla_x g(x) D^g, \quad (16.3.4)$$

where  $D^h$  and  $D^g$  are diagonal matrices such that

$$(D_{ii}^h + 1)^2 + (D_{ii}^g + 1)^2 \leq 1 \quad (i = 1, \dots, n). \quad (16.3.5)$$

If  $x \in \mathcal{D}$ , these matrices are then unique and given by

$$D_{ii}^h = \frac{h_i(x)}{\sqrt{h_i^2(x) + g_i^2(x)}} - 1 \text{ and } D_{ii}^g = \frac{g_i(x)}{\sqrt{h_i^2(x) + g_i^2(x)}} - 1, \quad (16.3.6)$$

in which case  $\Phi$  itself is differentiable at  $x$  with

$$\nabla_x \Phi(x) = V(x) = \nabla_x h(x) D^h + \nabla_x g(x) D^g.$$

We may now consider solving the system (16.3.3) in the least-squares sense, which leads us to consider the *complementarity merit function*

$$\Psi(x) = \frac{1}{2} \|\Phi(x)\|_2^2, \quad (16.3.7)$$

corresponding to (16.1.4). Remarkably, this function is itself continuously differentiable, with its gradient being given by

$$\nabla_x \Psi(x) = V(x)^T \Phi(x) \quad (16.3.8)$$

for all  $x \in \mathbb{R}^n$ . It therefore satisfies the assumption AF.1n that we made when considering nonsmooth unconstrained minimization problems in Chapter 11.

### 16.3.2 Applying the Basic Nonsmooth Algorithm

We have just seen that minimizing  $\Psi(x)$  is a problem to which our usual trust-region algorithms may, we hope, be applied. The first step is to decide which model of the objective function to use. A natural choice is a quadratic Gauss–Newton model for  $\Psi$  (see Section 16.1); that is,

$$\begin{aligned} m_k(x_k + s) &= \frac{1}{2} \|\Phi(x_k) + V(x_k)s\|_2^2 \\ &= \Psi(x_k) + \langle \nabla_x \Psi(x_k), s \rangle + \frac{1}{2} \langle s, V(x_k)^T V(x_k)s \rangle \\ &= \frac{1}{2} \|\Phi(x_k)\|_2^2 + \langle \Phi(x_k), V(x_k)s \rangle + \frac{1}{2} \|V(x_k)s\|_2^2. \end{aligned} \quad (16.3.9)$$

Of course, we need to compute  $V(x_k)$  for each iterate  $x_k$  for the model to be well defined. Given the Jacobian matrices of  $h$  and  $g$  at  $x_k$ , we must therefore compute the diagonal matrices  $D^h$  and  $D^g$  to satisfy (16.3.5). If  $h_i^2(x) + g_i^2(x) > 0$  for some  $i$ , (16.3.6) provides the values of  $D_{ii}^h$  and  $D_{ii}^g$ . Furthermore, these values are not needed if

$$h_i(x_k)^2 + g_i(x_k)^2 = 0 \text{ and } \|\nabla_x h_i(x_k)\|_2 + \|\nabla_x g_i(x_k)\|_2 = 0$$

since it follows from the second of these relations that the  $i$ th row of  $V(x_k)$  must be identically zero. The only remaining difficulty is therefore to compute  $D_{ii}^h$  and  $D_{ii}^g$  in the case where

$$h_i(x_k)^2 + g_i(x_k)^2 = 0 \text{ and } \|\nabla_x h_i(x_k)\|_2 + \|\nabla_x g_i(x_k)\|_2 > 0.$$

Let  $\mathcal{I}_k$  be the set of indices in  $\{1, \dots, n\}$  for which these last two conditions hold, and assume that it is not empty. Without loss of generality, we may suppose that  $\mathcal{I}_k = \{1, \dots, p\}$  for some  $p \leq n$ , and also that  $\|\nabla_x h_i(x_k)\|_2 \neq 0$  for each  $i \in \mathcal{I}_k$ . If we assume that we can compute a nonzero direction  $d_k$  such that

$$\langle \nabla_x h_i(x_k), d_k \rangle \neq 0 \text{ for } i \in \mathcal{I}_k, \quad (16.3.10)$$

we then define, for small positive  $t$ ,

$$y_k(t) \stackrel{\text{def}}{=} x_k + t d_k + w(t),$$

where  $w(t)$  is chosen such that  $\|w(t)\|_2$  goes to zero faster than  $t$  and  $\Phi$  is differentiable at  $y_k(t)$ , the latter requirement being possible because of Rademacher's theorem (see Section 3.1.2) and the Lipschitz continuity of  $\Phi$ . Letting  $t$  tend to zero and using Clarke's theorem (see Section 3.1.4), we obtain from (16.3.6) that, for  $i \in \mathcal{I}_k$ ,

$$D_{ii}^h = \frac{\langle \nabla_x h_i(x), d_k \rangle}{\sqrt{\langle \nabla_x h_i(x), d_k \rangle^2 + \langle \nabla_x g_i(x), d_k \rangle^2}} - 1$$

and

$$D_{ii}^g = \frac{\langle \nabla_x g_i(x), d_k \rangle}{\sqrt{\langle \nabla_x h_i(x), d_k \rangle^2 + \langle \nabla_x g_i(x), d_k \rangle^2}} - 1,$$

and we are done. The matrix  $V(x_k)$  is thus well defined, as is the gradient (16.3.8) and the model (16.3.9).

But how do we compute nonzero  $d_k$  such that (16.3.10) holds? Our technique is as follows. We first set  $d_k = \nabla_x h_1(x_k)$  and define

$$\mathcal{J}_k = \{i \in \mathcal{I}_k \mid \langle \nabla_x h_i(x_k), d_k \rangle = 0\}.$$

If  $\mathcal{J}_k$  is empty, then  $d_k$  satisfies (16.3.10). Otherwise, we may choose an arbitrary  $j \in \mathcal{J}_k$  and define

$$\hat{d}_k = d_k + \frac{\min_{i \in \mathcal{I}_k \setminus \mathcal{J}_k} |\langle \nabla_x h_i(x_k), d_k \rangle|}{2 \max_{i \in \mathcal{J}_k} \|\nabla_x h_i(x_k)\|_2 \|\nabla_x h_j(x_k)\|_2} \nabla_x h_j(x_k).$$

By definition, we then obtain that  $\langle \nabla_x h_j(x_k), \hat{d}_k \rangle \neq 0$ . Moreover, we have that

$$\begin{aligned} \langle \nabla_x h_i(x_k), \hat{d}_k \rangle &= \langle \nabla_x h_i(x_k), d_k \rangle \\ &+ \frac{1}{2} \min_{i \in \mathcal{I}_k \setminus \mathcal{J}_k} |\langle \nabla_x h_i(x_k), d_k \rangle| \frac{\langle \nabla_x h_i(x_k), \nabla_x h_j(x_k) \rangle}{\max_{i \in \mathcal{J}_k} \|\nabla_x h_i(x_k)\|_2 \|\nabla_x h_j(x_k)\|_2} \end{aligned}$$

for  $i \in \mathcal{I}_k \setminus \mathcal{J}_k$ . The Cauchy–Schwarz inequality then implies that  $\langle \nabla_x h_i(x_k), \hat{d}_k \rangle \neq 0$  for such  $i$ . If

$$\hat{\mathcal{J}}_k \stackrel{\text{def}}{=} \{i \in \mathcal{I}_k \mid \langle \nabla_x h_i(x), \hat{d}_k \rangle = 0\} \subseteq \mathcal{J}_k \setminus \{j\}$$

is the empty set, then  $\hat{d}_k$  satisfies (16.3.10). Otherwise, we redefine  $\mathcal{J}_k \leftarrow \hat{\mathcal{J}}_k$ ,  $d_k \leftarrow \hat{d}_k$ , select a new index in  $\mathcal{J}_k$ , and repeat the procedure. After at most  $p$  steps, we must obtain that  $\mathcal{J}_k = \emptyset$ , and thus the corresponding  $d_k$  satisfies (16.3.10), as desired.

Observe that the model (16.3.9) involves  $\nabla_x h(x)$  and  $\nabla_x g(x)$  as parameters (via the definition of  $V(x)$ ). For the model to satisfy AM.4n it thus suffices to assume that these Jacobian matrices stay bounded, since we immediately see that (16.3.5) ensures boundedness of  $D^h$  and  $D^g$ .

**AC.5e** There exists a constant<sup>277</sup>  $\kappa_{\text{cmp}} > 0$  such that

$$\|\nabla_x h(x)\|_2 \leq \kappa_{\text{cmp}} \quad \text{and} \quad \|\nabla_x g(x)\|_2 \leq \kappa_{\text{cmp}}$$

for all  $x \in \mathbb{R}^n$ .

This assumption is reasonable, as it ensures that the curvature of the model (16.3.9) cannot be arbitrarily large, which might otherwise lead to arbitrarily small steps. In addition, the model clearly satisfies assumptions AM.1n–AM.3n. As a consequence, we only need to verify that we can choose a step  $s_k$  for which AA.1n holds in order to apply our nonsmooth variant of the basic algorithm. But this last requirement is clearly fulfilled since (16.3.9) is a smooth quadratic, to which the results of Section 6.3 obviously apply. We therefore conclude that, under assumption AC.5e, *the nonsmooth basic trust-region algorithms (Algorithm 11.1.1 [p. 413], its nonmonotone counterpart, and the variant, Algorithm 11.3.1 [p. 425]) converge globally for the problem of minimizing the complementarity merit function (16.3.7), in the sense of Theorems 11.2.5 (p. 420) and 11.3.2 (p. 426).*

### 16.3.3 Regular Complementarity Problems and Their Solutions

Once we have found a first-order critical point for the complementarity merit function, we still need to relate such a point to a solution of the original problem. In order to develop this formally, we define the following index sets:

$$\mathcal{J}_C(x) \stackrel{\text{def}}{=} \{i \in \{1, \dots, n\} \mid h_i(x) \geq 0, g_i(x) \geq 0, \text{ and } h_i(x)g_i(x) = 0\},$$

$$\mathcal{J}_+(x) = \{i \in \{1, \dots, n\} \mid h_i(x) > 0 \text{ and } g_i(x) > 0\},$$

---

<sup>277</sup>“cmp” stands for “complementarity”.

and

$$\mathcal{J}_-(x) = \{i \in \{1, \dots, n\} \setminus (\mathcal{J}_C(x) \cup \mathcal{J}_+(x))\}.$$

Note that  $\mathcal{J}_C(x) \cap \mathcal{J}_+(x) = \emptyset$ , and therefore that the three index sets form a partition of  $\{1, \dots, n\}$ . We may then define the generalized complementarity problem to be *regular* at the point  $x$  if, for any two nonzero vectors  $v_1$  and  $v_2$  of  $\mathbb{R}^n$  such that (componentwise)

$$[v_i]_{\mathcal{J}_C(x)} = 0, \quad [v_i]_{\mathcal{J}_+(x)} > 0, \quad [v_i]_{\mathcal{J}_-(x)} < 0 \quad (i = 1, 2),$$

we have that

$$\nabla_x h(x)v_1 + \nabla_x g(x)v_2 \neq 0.$$

We may then prove the following result.

**Theorem 16.3.1** A vector  $x$  is a solution of the generalized complementarity problem if and only if this problem is regular at  $x$  and  $x$  is a first-order critical point of the function  $\Psi(x)$ .

**Proof.** Suppose first that  $x$  is a solution of the generalized complementarity problem. Then it must be a first-order critical point of  $\Psi(x)$  (with  $\Psi(x) = 0$ ). Hence  $\mathcal{J}_+(x)$  and  $\mathcal{J}_-(x)$  must be empty, which then implies, by definition, that the problem is regular at  $x$ .

Conversely, suppose that  $x$  is a regular first-order critical point of  $\Psi(x)$  and let  $D^h$  and  $D^g$  be diagonal matrices satisfying (16.3.5) and (16.3.6). Let

$$v_1 = D^h \Phi(x) \text{ and } v_2 = D^g \Phi(x).$$

Then we have, from (16.3.8) and (16.3.4), that

$$0 = \nabla_x \Psi(x) = \nabla_x h(x)D^h \Phi(x) + \nabla_x g(x)D^g \Phi(x) = \nabla_x h(x)v_1 + \nabla_x g(x)v_2. \quad (16.3.11)$$

If  $x$  does not solve the generalized complementarity problem, then we first note that  $\mathcal{J}_+(x) \cup \mathcal{J}_-(x)$  cannot be empty. Consider first an index  $i \in \mathcal{J}_+(x)$ . Then, by definition of this set,  $h_i^2(x) + g_i^2(x) > 0$  and (16.3.6) holds with

$$D_{ii}^h < 0 \text{ and } D_{ii}^g < 0. \quad (16.3.12)$$

From the triangle inequality, we also obtain that  $\phi(h_i(x), g_i(x)) < 0$ , which then implies, with (16.3.12), that for  $i \in \mathcal{J}_+(x)$ ,

$$[v_1]_i = D_{ii}^h \phi(h_i(x), g_i(x)) > 0 \text{ and } [v_2]_i = D_{ii}^g \phi(h_i(x), g_i(x)) > 0.$$

If, on the other hand,  $i \in \mathcal{J}_-(x)$ , then either  $h_i(x) < 0$  or  $g_i(x) < 0$ , or both. This again implies that  $h_i^2(x) + g_i^2(x) > 0$ , and therefore that (16.3.6) holds. Moreover, we

immediately obtain that (16.3.12) also holds in this case and that  $\phi(h_i(x), g_i(x)) > 0$ . Consequently, we obtain that for  $i \in \mathcal{J}_-(x)$ ,

$$[v_1]_i = D_{ii}^h \phi(h_i(x), g_i(x)) < 0 \text{ and } [v_2]_i = D_{ii}^g \phi(h_i(x), g_i(x)) < 0.$$

Thus  $v_1 \neq 0 \neq v_2$ . The regularity assumption then gives that

$$\nabla_x h(x)v_1 + \nabla_x g(x)v_2 \neq 0,$$

which contradicts (16.3.11). Hence  $x$  must be a solution of the generalized complementarity problem.  $\square$

There are conditions on the Jacobian matrices  $\nabla_x h(x)$  and  $\nabla_x g(x)$  under which the problem can be proved to be regular at  $x$ . For instance, this is the case for a *monotone* NCP whose Jacobian is positive definite (although possibly nonsymmetric). More general conditions do not appear to provide more geometric intuition than the definition of regularity itself, and we do not explore this avenue further.

## Notes and References for Section 16.3

A number of Newton-type methods (based on the system (16.3.3)) have been designed for the solution of the NCP, as the surveys of Harker and Pang (1990) and Pang (1995) make clear. In most of these methods, global convergence is ensured within a linesearch framework. Other interesting references include Harker and Xiao (1990), Zhang (1994), De Luca, Facchinei, and Kanzow (1995), Facchinei, Fischer, and Kanzow (1997), and Jiang and Qi (1997).

The exposition of trust-region methods given here was inspired by the work of Zupke (1997), Kanzow and Zupke (1998), and Jiang et al. (1998). The first two of these references allow for an inexact solution of the trust-region subproblem and also include results on the rate of convergence of the proposed algorithms as well as providing additional sufficient conditions to ensure the regularity of the problem. They also contain a special trust-region radius updating technique that ensures a minimal (constant) radius after each successful iteration, although this technique does not appear to be necessary to obtain global convergence to first-order critical points. We also recommend the paper by Facchinei and Soares (1997) as a clear source of information on the Fischer–Burmeister function and an associated algorithm. The notion of regularity that we used is due to Robinson (1983).

The design of complementarity merit function appears to be a very active area of research. The Fischer–Burmeister function, with its advantageous differentiability properties, was introduced by Fischer (1992). Its use in various context is described in the survey by Fischer (1995). Other complementarity merit functions and the algorithms that can be built by using them have been studied Fukushima (1992), Mangasarian and Solodov (1993), Yamashita and Fukushima (1995), Peng (1996, 1998), Tseng, Yamashita, and Fukushima (1996), Chen, Chen, and Kanzow (1997), Kanzow and Zupke (1998), Facchinei and Kanzow (1997), Li (1997), Luo and Tseng (1997), and Kanzow, Yamashita, and Fukushima (1997), among others. A relatively recent survey on this area of research is given by Fukushima (1996).

Complementarity problems are not limited to the classes we have presented. We refer the reader to De Schutter and De Moor (1997) for extensions of the LCP and to Gabriel

and Moré (1995), Chen and Mangasarian (1996), Billups and Ferris (1997), and Ferris, Kanzow, and Munson (1999) for the solution of *mixed complementarity problems*, which are obtained by imposing upper bounds on the variables of an NCP. The PATH package by Dirkse and Ferris (1995) is high-quality software for the numerical solution of such problems. An interesting trust-region method for bound-constrained semismooth equations with applications to the nonlinear mixed complementarity problem is that of Ulbrich (1999), where yet another complementarity function is introduced that is then minimized subject to the bounds of the variables. It is also important to notice that mixed complementarity problems can be viewed as a special case of general *variational inequalities* where, for a given function  $g$  from  $\mathbb{R}^n$  into itself and a closed convex set  $X \subseteq \mathbb{R}^n$ , one seeks a vector  $x \in X$  such that

$$\langle g(x), y - x \rangle \geq 0 \text{ for all } y \in X. \quad (16.3.13)$$

This connects complementarity problems with a vast area of research including mathematical programs with equilibrium constraints (MPECs), in which case (16.3.13) describes an equilibrium situation for some system. This in turn leads to a large number of applications, ranging from traffic assignment problems (see Dafermos, 1980; Sheffi, 1985; Patriksson, 1994; or Colson, 1999) to general economics (see, for instance, Nagurney, 1993). Patriksson (1998) proposes a unifying view of this connection in a linesearch-based framework. Note that some of the complementarity merit functions considered in the references cited above are actually applicable in this more general context. A trust-region method for solving general nonlinearly constrained variational inequality problems is discussed by Yang, Li, and Zhou (1998), while a method applicable to MPECs is discussed by Stöhr (1999) and Scholtes and Stöhr (1999). Engineering and economic applications of this research area are surveyed by Ferris and Pang (1997a). Finally, we refer the interested reader to the recent books edited by Ferris and Pang (1997b) and Fukushima and Qi (1998) for further issues and developments in the domains of complementarity problems and variational inequalities.

## Part V

---

# Final Considerations

In this part of the book, we examine some points that remain. In particular, we discuss practical issues arising in the implementation of trust-region methods, try to put what we have covered into perspective, and, finally, provide pointers to all of the material we have developed before.

---

# Chapter 17

---

## Practicalities

---

Having finished our analysis of trust-region methods and their convergence properties, we return, at last, to a discussion of issues that were not covered in detail in our previous chapters but that may nevertheless contribute significantly to the practical success (or failure) of trust-region algorithms.

### 17.1 Choosing the Algorithmic Parameters

Algorithm BTR, like other trust-region algorithms covered in this book, depends on a set of parameters. Here, we continue our discussion, which we started in the previous chapters, of what we have found to be appropriate values for these parameters. We start by considering the two parameters  $\eta_1$  and  $\eta_2$ , which define the ranges of values for  $\rho_k$ , which is in turn the ratio of achieved to predicted reduction, for which a trial step is judged successful or not (this is the role of  $\eta_1$ ) and, if successful, whether very successful or not (this is the role of  $\eta_2$ ). In the notes for Section 6.4, we indicated that choosing  $\eta_1$  to be strictly positive or zero has some influence on the theoretical properties of the basic algorithm, but we also pointed out that the difference between a zero value of this parameter and a strictly positive but small value does not make a large difference in algorithmic behaviour. This prompted us to recommend  $\eta_1 > 0$  in order to ensure the best possible convergence guarantees. We now consider how large  $\eta_1$  and  $\eta_2$  should be, subject to the constraint  $0 < \eta_1 \leq \eta_2 < 1$ . As far as we are aware, the only approach to this question is experimental. From our experience, values of

$$\eta_1 = 0.05 \text{ and } \eta_2 = 0.9 \quad (17.1.1)$$

appear to give good performance on average, when performance is measured either with respect to number of iterations required or in terms of computing time. Figure 17.1.1 shows the (normalized) average number of iterations required for convergence<sup>278</sup> of the basic algorithm on 26 difficult unconstrained problems with 1,000 variables from the CUTE collection.

---

<sup>278</sup>With the choice (17.1.2).

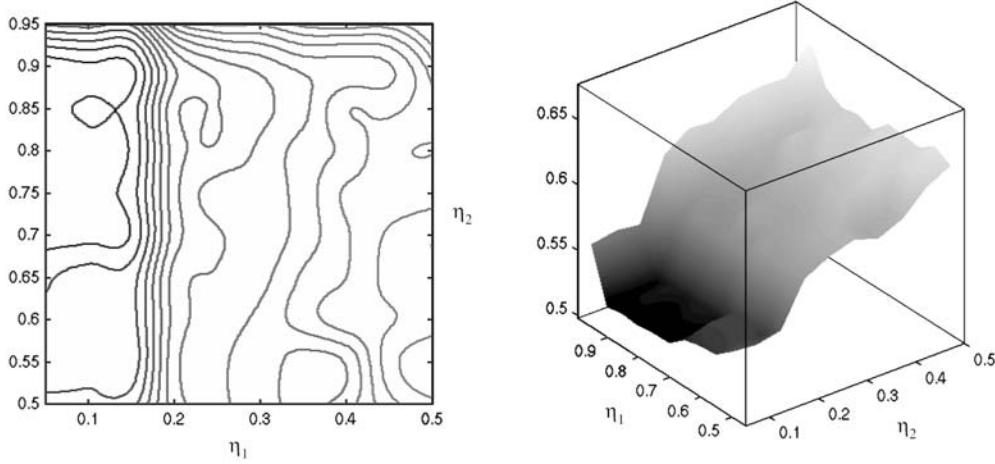


Figure 17.1.1: Average number of iterations of the basic algorithm for 26 medium-size problems as a function of  $\eta_1$  and  $\eta_2$ .

These numbers therefore suggest that the criterion for accepting a trial step should be quite weak, which is in line with our earlier theoretical discussion. We also note that the suggested values for  $\eta_2$  indicate that it is better, in practice, to increase the trust-region radius only for values of  $\rho_k$  that are quite close to, or larger than, 1.

As we already discussed in Section 10.5.2, the details of the procedure used to update the trust-region radius may have a crucial influence on the overall performance of a trust-region method. In that section, we argued that a trust-region radius updating rule of the form

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k) & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \|s_k\|, \gamma_2 \|s_k\|] & \text{if } \rho_k < \eta_1, \end{cases}$$

where

$$0 < \gamma_1 \leq \gamma_2 \leq 1,$$

is useful in practice. This still leaves considerable flexibility. A number of practical implementations consider the simple choice

$$\Delta_{k+1} \in \begin{cases} \max[\alpha_1 \|s_k\|_k, \Delta_k] & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \alpha_2 \|s_k\|_k & \text{if } \rho_k < \eta_1 \end{cases}$$

for some constants  $\alpha_1$  and  $\alpha_2$  satisfying the inequality  $0 < \alpha_2 < 1 \leq \alpha_1$ . In this framework, numerical testing supports the view that values

$$\alpha_1 = 2.5 \text{ and } \alpha_2 = 0.25 \tag{17.1.2}$$

give good average performance. Figure 17.1.2 shows the (normalized) average number

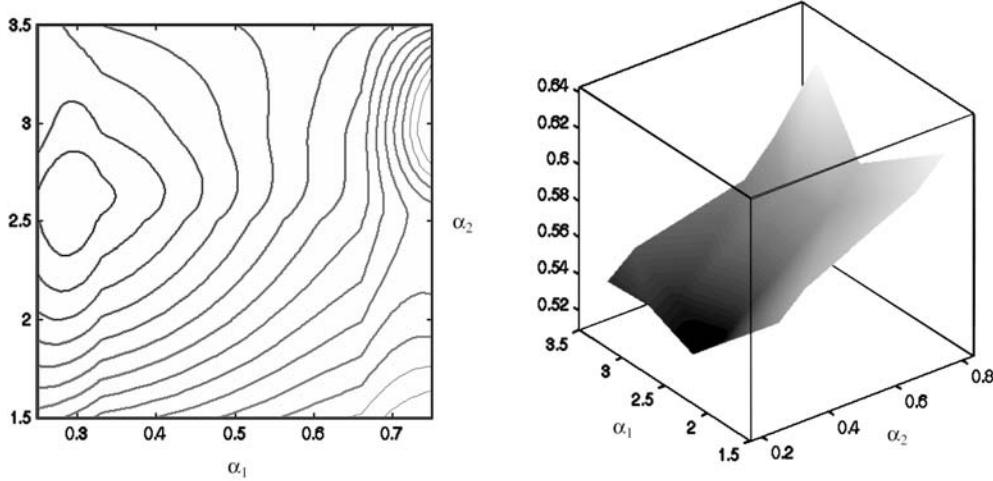


Figure 17.1.2: Average number of iterations of the basic algorithm for 26 medium-size problems as a function of  $\alpha_1$  and  $\alpha_2$ .

of iterations<sup>279</sup> required by the basic algorithm to converge on the same collection of 26 CUTE problems that was already considered in Figure 17.1.1.

But further refinements are possible. In particular, if the agreement between the model and the function is so poor that  $\rho_k < 0$ , some drastic action might be in order. In this case, we might consider just how large a radius would need to be to allow us to make a very successful step if the step taken were in the direction we have just attempted to move and if the true function were a quadratic. A simple interpolation calculation reveals that, so long as AM.2 and AM.3 hold, we should then choose

$$\Delta_{k+1} = \max[\gamma_1, \gamma_{\text{bad}}] \Delta_k,$$

where

$$\gamma_{\text{bad}} = \frac{(1 - \eta_2)\langle g_k, s_k \rangle}{(1 - \eta_2)[f_k + \langle g_k, s_k \rangle + \eta_2 m_k(x_k + s_k) - f(x_k + s_k)]}. \quad (17.1.3)$$

We have used a value of  $\gamma_1 = 0.0625$  with some success in the past. To summarize, the rule

$$\Delta_{k+1} = \begin{cases} \max[\alpha_1 \|s_k\|_k, \Delta_k] & \text{if } \rho_k \geq \eta_2, \\ \Delta_k & \text{if } \rho_k \in [\eta_1, \eta_2), \\ \alpha_2 \|s_k\|_k & \text{if } \rho_k \in [0, \eta_1), \\ \min[\alpha_2 \|s_k\|_k, \max[0.0625, \gamma_{\text{bad}}] \Delta_k] & \text{if } \rho_k < 0, \end{cases} \quad (17.1.4)$$

where  $\gamma_{\text{bad}}$  is given by (17.1.3), seems appropriate when AM.2 and AM.3 hold.

## Notes and References for Section 17.1

The discussion of this section is based on the experience we have gained with LANCELOT (see

<sup>279</sup>With the choice (17.1.1).

Conn, Gould, and Toint, 1992b, p. 116), and the experimental study of Gould, Orban, Sartenaer, and Toint (2000b), which appears to be the only systematic investigation in this direction. Figures 17.1.1 and 17.1.2 are extracted from this latter source. Interpolation/extrapolation techniques for the radius have been considered by Dennis and Schnabel (1983, Section 6.4.3) and Auer (1993). It is interesting that the values given substantially differ from the values  $\eta_1 = 0.25$ ,  $\eta_2 = 0.75$ ,  $\alpha_1 = 0.5$ , and  $\alpha_2 = 2$  that have often been quoted in the literature (see, for instance, Dennis and Mei, 1979; Conn, Gould, and Toint, 1988b; Sebudiandi and Toint, 1993; Dennis and Vicente, 1996; Coleman and Li, 1996a; Shahabuddin, 1996; Dennis and Torczon, 1997; Diniz-Ehrhardt, Gomes-Ruggiero, and Santos, 1998; or Dennis, Heinkenschloss, and Vicente, 1998). The performance of the basic algorithm using the values of the parameters recommended in (17.1.1) and (17.1.2) appears to be nearly twice as good as that obtained with the more commonly used values. Further experimental work is necessary to appraise the real power of the interpolation formula (17.1.3) and, more generally, of updating rules like (17.1.4).

## 17.2 Choosing the Initial Trust-Region Radius

Another significant issue is the choice of the initial trust-region radius  $\Delta_0$ . In all the algorithms described so far, including Algorithm BTR, we have assumed that an initial  $\Delta_0$  is given. This implicitly assumes that users typically have some idea of the size of the region where the initial model  $m_0$  is adequate. Unfortunately, this optimistic view is seldom justified in our experience, and one very often has to resort to some heuristic to choose  $\Delta_0$  on the basis of other initial information. The purpose of this section is to review some of the techniques that have been proposed for this task. For simplicity, we focus on the unconstrained case.

A few simple strategies have been suggested and used in the past, such as choosing

$$\Delta_0 = 1 \text{ or } \Delta_0 = \frac{1}{10} \|g_0\|, \quad (17.2.1)$$

and we note that they appear to be numerically reasonable and acceptably efficient if the problems considered are well scaled. A more elaborate technique, taking the size of the model's Hessian at  $x_0$  into account, is to choose the radius as the length of model minimizer in the steepest-descent direction when it exists; that is,

$$\Delta_0 = \frac{\|g_0\|^2}{\langle g_0, \nabla_{xx} m_0(x_0) g_0 \rangle}. \quad (17.2.2)$$

This amounts to ensuring that the Cauchy point lies on the boundary of the trust region. Of course, when  $\langle g_0, \nabla_{xx} m_0(x_0) g_0 \rangle \leq 0$ , this is not practical, and one could think of using

$$\Delta_0 = \frac{\|g_0\|^2}{|\langle g_0, \nabla_{xx} m_0(x_0) g_0 \rangle|}$$

instead.

The idea of examining how far the model can be trusted along the steepest-descent direction is interesting because its cost in terms of linear algebra is minimal, and it can be explored further. The basic idea in the algorithm that we now describe is to determine the largest possible radius such that the model prediction is sufficiently close to the true objective function value (along the steepest descent), while at the same time exploiting any improvement obtained in the objective function itself during the process.

The procedure is iterative, and its iterations will be indexed by bracketed superscripts. Thus we assume that we start with the initial point  $x_0$  and initial (arbitrary) trust-region radius  $\Delta_0^{(0)}$ ,  $f_0 = f(x_0)$  and  $g_0 = \nabla_x m_0(x_0)$ . At iteration  $i$  of the procedure, we compute

$$d_0^{(i)} = \frac{\Delta_0^{(i)}}{\|g_0\|} g_0, \quad m_0^{(i)} = m_0(x_0 - d_0^{(i)}), \quad \text{and} \quad f_0^{(i)} = f(x_0 - d_0^{(i)}) \quad (17.2.3)$$

as well as the ratio

$$\rho_0^{(i)} \stackrel{\text{def}}{=} \frac{f_0 - f_0^{(i)}}{f_0 - m_0^{(i)}}. \quad (17.2.4)$$

We will say that the model predicts the objective badly for the radius  $\Delta_0^{(i)}$  when

$$|\rho_0^{(i)} - 1| > \mu_1 \quad (17.2.5)$$

or that it predicts it well when

$$|\rho_0^{(i)} - 1| \leq \mu_2 \quad (17.2.6)$$

for some constants  $0 \leq \mu_2 < \mu_1$ . We then consider trying a new value for the radius given by

$$\Delta_0^{(i+1)} = \tau^{(i)} \Delta_0^{(i)} \quad \text{with} \quad \tau^{(i)} \in \begin{cases} [\bar{\gamma}_1, 1] & \text{if (17.2.5) holds,} \\ [1, \bar{\gamma}_4] & \text{if (17.2.6) holds,} \\ [\bar{\gamma}_2, \bar{\gamma}_3] & \text{otherwise} \end{cases} \quad (17.2.7)$$

for some  $0 < \bar{\gamma}_1 \leq \bar{\gamma}_2 \leq 1 \leq \bar{\gamma}_3 \leq \bar{\gamma}_4$ .<sup>280</sup> The fact that we have chosen  $\tau^{(i)}$  in prescribed intervals then allows us to use the remaining freedom (within those intervals) to further improve our estimate, which we can do by using quadratic interpolation for the univariate function  $f(x_0 - \tau d_0^{(i)})$ . We therefore define

$$q^{(i)}(\tau) \stackrel{\text{def}}{=} f_0 - \tau \langle g_0, d_0^{(i)} \rangle + \tau^2 (f_0^{(i)} - f_0 + \langle g_0, d_0^{(i)} \rangle)$$

and use this quadratic to determine new values of  $\tau^{(i)}$  for which the ratio  $\rho_0^{(i)}$  would be close to 1, in the sense that

$$\frac{f_0 - q^{(i)}(\tau)}{f_0 - m_0(x_0 - \tau d_0^{(i)})} = 1 - \theta \quad \text{or} \quad \frac{f_0 - q^{(i)}(\tau)}{f_0 - m_0(x_0 - \tau d_0^{(i)})} = 1 + \theta$$

---

<sup>280</sup>We have chosen to overline these constants to distinguish them from the constants used in the trust-region update at later iterations (see Step 4 of Algorithm BTR [p. 116]).

for some  $\theta \in (0, \mu_2]$ . We therefore solve these two equations for  $\tau$ , which yields the roots

$$\tau_1^{(i)} = \frac{-\theta \langle g_0, d_0^{(i)} \rangle}{\theta(f_0 - \langle g_0, d_0^{(i)} \rangle) + (1 - \theta)m_0^{(i)} - f_0^{(i)}} \quad (17.2.8)$$

and

$$\tau_2^{(i)} = \frac{\theta \langle g_0, d_0^{(i)} \rangle}{-\theta(f_0 - \langle g_0, d_0^{(i)} \rangle) + (1 + \theta)m_0^{(i)} - f_0^{(i)}}, \quad (17.2.9)$$

as well as

$$\tau_{\min} = \min[\tau_1^{(i)}, \tau_2^{(i)}] \text{ and } \tau_{\max} = \max[\tau_1^{(i)}, \tau_2^{(i)}]. \quad (17.2.10)$$

We now reconsider the three cases arising in the second part of (17.2.7). In all cases, our policy is to give priority to *relevant* values of  $\tau$ , that is, values smaller than 1 if the radius has to be decreased, or larger than 1 if it has to be increased.<sup>281</sup> We also wish to give greater weight to the *largest* of the relevant values of  $\tau$ . Keeping this strategy in mind, we start by assuming that we consider the first of the three cases, that is, when  $\tau^{(i)}$  must be chosen in the interval  $[\bar{\gamma}_1, 1)$ , which is when the radius should be decreased. We then select

$$\tau^{(i)} = \begin{cases} \bar{\gamma}_2 & \text{if } \tau_{\min} > 1, \\ \bar{\gamma}_1 & \text{if } \tau_{\max} < \bar{\gamma}_1 \text{ or} \\ & \text{if } \tau_{\min} < \bar{\gamma}_1 \text{ and } \tau_{\max} \geq 1, \\ \tau_1^{(i)} & \text{if } \tau_1^{(i)} \in [\bar{\gamma}_1, 1) \text{ and } \tau_2^{(i)} \notin [\bar{\gamma}_1, 1), \\ \tau_2^{(i)} & \text{if } \tau_2^{(i)} \in [\bar{\gamma}_1, 1) \text{ and } \tau_1^{(i)} \notin [\bar{\gamma}_1, 1), \\ \tau_{\max} & \text{otherwise.} \end{cases} \quad (17.2.11)$$

In the first of these cases, none of the values given by (17.2.8)–(17.2.9) is relevant and we choose a fixed value,  $\bar{\gamma}_2$ , in the desired interval. We choose  $\bar{\gamma}_1$ , the lower end of the interval, when the relevant information ( $\tau_{\min}$ ) suggests a reduction that is larger than allowed by the chosen interval. The third and fourth cases give preference to the relevant value of  $\tau$  in the interval, while the last case simply chooses the largest of two acceptable and relevant values. Applying the same type of reasoning in the second case of (17.2.7), that is, when  $\tau$  should be increased by selecting it from the interval  $[1, \bar{\gamma}_4]$ , and giving preference to large values again, we obtain that

$$\tau^{(i)} = \begin{cases} \bar{\gamma}_3 & \text{if } \tau_{\max} < 1, \\ \bar{\gamma}_4 & \text{if } \tau_{\max} > \bar{\gamma}_4, \\ \tau_1^{(i)} & \text{if } \tau_1^{(i)} \in [1, \bar{\gamma}_4] \text{ and } \tau_2^{(i)} < 1, \\ \tau_2^{(i)} & \text{if } \tau_2^{(i)} \in [1, \bar{\gamma}_4] \text{ and } \tau_1^{(i)} < 1, \\ \tau_{\max} & \text{otherwise.} \end{cases} \quad (17.2.12)$$

Similarly, we obtain in the last case that

$$\tau^{(i)} = \begin{cases} \bar{\gamma}_2 & \text{if } \tau_{\max} < \bar{\gamma}_2, \\ \bar{\gamma}_3 & \text{if } \tau_{\max} > \bar{\gamma}_3, \\ \tau_{\max} & \text{otherwise.} \end{cases} \quad (17.2.13)$$

---

<sup>281</sup>This is important because of the somewhat hazardous nature of extrapolation.

Note that in this last case, (17.2.7) does not state whether the radius should be increased or decreased, which is why we decide to always prefer the maximum of the two possible values for  $\tau$ .

Given this somewhat intricate safeguarded interpolation/extrapolation scheme, we are now in position to state the complete algorithm for determining the initial trust-region radius. In this algorithm, we maintain a value  $\Delta^{\max}$  which is the maximal radius for which an adequate model prediction was found in the course of the algorithm. We allow, as we mentioned in our main objectives, any progress in the objective function to be taken into account by accepting to redefine (in Step 3) the current “initial point”  $x_0$  if another point is found with a lower objective function value. We also limit the amount of searching and starting point redefinitions by imposing upper bounds  $i_{\max}$  and  $j_{\max}$  on the number of allowed trial radii for a given starting point and on the number of starting points, respectively.

**Algorithm 17.2.1: Choice of the initial trust-region radius**

**Step 0: Initialization.** An initial point  $x_0$  and an (arbitrary) initial  $\Delta_0^{(0)}$  are given. Compute  $f_0$  and  $g_0$ . Then set  $i = j = 0$ ,  $f_{\min} = f_0$ , and  $\Delta_0^{\max} = 0$ .

**Step 1: Update of the maximal radius.** Compute  $d_0^{(i)}$ ,  $m_0^{(i)}$ ,  $f_0^{(i)}$ , and  $\rho_0^{(i)}$  according to (17.2.3) and (17.2.4). If

$$|\rho_0^{(i)} - 1| \leq \mu_1,$$

set  $\Delta_0^{\max} = \max[\Delta_0^{\max}, \Delta_0^{(i)}]$ .

Moreover, if  $j \leq j_{\max}$  and  $f_0^{(i)} < f_{\min}$ , redefine

$$f_{\min} \leftarrow f_0^{(i)} \text{ and } \sigma \leftarrow \frac{\Delta_0^{(i)}}{\|g_0\|}.$$

**Step 2: Update the radius.** If  $i > i_{\max}$ , go to Step 3. Otherwise, compute the values of (17.2.8)–(17.2.10), set

$$\tau^{(i)} \left\{ \begin{array}{l} \text{using (17.2.11) if } |\rho_0^{(i)} - 1| > \mu_1, \\ \text{using (17.2.12) if } |\rho_0^{(i)} - 1| \leq \mu_2, \\ \text{using (17.2.13) otherwise,} \end{array} \right.$$

and  $\Delta_0^{(i+1)} = \tau^{(i)} \Delta_0^{(i)}$ . Increment  $i$  by 1 and return to Step 1.

**Step 3: Final update.** If  $j \leq j_{\max}$  and  $f_{\min} < f_0$ , redefine

$$x_0 \leftarrow x_0 - \sigma g_0,$$

increment  $j$  by 1, and return to Step 0. Else, set  $\Delta_0 = \max(\Delta_0^{(i)}, \Delta_0^{\max})$  and stop.

Note that the bounds  $i_{\max}$  and  $j_{\max}$  together limit the number of additional evaluations of the objective function in the complete procedure to a maximum of  $(i_{\max} + 1)(j_{\max} + 1)$ .

In practice, the constants appearing in the algorithm must be specified. The values

$$\begin{aligned}\bar{\gamma}_1 &= 0.0625, & \bar{\gamma}_2 &= 0.5, & \bar{\gamma}_3 &= 2, & \bar{\gamma}_4 &= 5, \\ \mu_1 &= 0.5, & \mu_2 &= 0.35, & \theta &= 0.25, & i_{\max} &= 4, \text{ and } j_{\max} = 1\end{aligned}\quad (17.2.14)$$

are reported to give significant improvements in average algorithm performance over the simpler strategies mentioned at the beginning of this section.

Another interesting possibility is to use internal doubling (see Section 10.5.1), at least on the very first iteration of the algorithm. This procedure is potentially more expensive in terms of linear algebra, but has the merit of measuring the adequacy of the model along the direction where it really matters (the step) instead of measuring it along the steepest-descent direction. On the other hand, it may be wasteful in terms of linear algebra when the radius chosen (maybe using one of the other techniques mentioned above) to start the iteration is too large, at least in a straightforward implementation of the idea. A further sophistication is the use of internal doubling coupled with the backtracking technique (see Section 10.3.2), in order to avoid the possibly expensive recomputation of a shorter step if the first radius considered by the internal doubling procedure turns out to be too large.

We conclude this discussion by noting that software packages for which the user may specify if the problem at hand is linear or quadratic may sometimes have a considerable computational advantage when quadratic models with exact derivatives are used in the underlying algorithm. Indeed, the model and objective function coincide in this case, and the initial trust-region radius should therefore ideally be infinite. By contrast, if the initial radius is small compared to the distance from the starting point to the solution of the problem, several wasteful iterations may be required, in which the same model is successively minimized within increasingly large, but unfortunately not large enough, trust regions.

## Notes and References for Section 17.2

To our knowledge, the study of Sartenaer (1997), from which Algorithm 17.2.1 is extracted, is the only published source of comparative information on the choice of the initial trust-region radius. Sartenaer also proposed the particular values (17.2.14). The formulae (17.2.1) are used, for instance, within the **LANCELOT** package (see the discussion of the numerical performance of this package in Conn, Gould, and Toint, 1992a, 1996). The strategy (17.2.2) was suggested by Powell (1970b) in the context of a quadratic quasi-Newton method, where the choice of the initial model's Hessian may be poor. The results of Sartenaer (1997) indicate that (17.2.2) is comparable to the simpler, second choice in (17.2.1) when used in conjunction with a quadratic model with exact first and second derivatives.

### 17.3 Computing the Generalized Cauchy Point

For many of the methods we have considered, it is reassuring to compute the (generalized) Cauchy point, since then we can guarantee convergence of our specific method simply by reducing our model by at least a fixed fraction of the value attained at this point. When there are simple bounds on the variables,  $\ell \leq x \leq u$  (see Chapter 12), it is often convenient to use an  $\ell_\infty$ -norm trust region, since then the feasible region for the model problem is the “box”

$$\mathcal{B} = \{s \mid \ell - x \leq s \leq u - x \text{ and } -\Delta e \leq s \leq \Delta e\}.$$

Suppose that we intend to model our objective function  $f(x)$  by the quadratic approximation

$$q(s) = f + \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle. \quad (17.3.1)$$

In this case, a suitable generalized Cauchy point (see Section 12.2.1) is the first local minimizer of  $q(s(t))$  as a function of the scalar  $t$ , where

$$s(t) = P_{\mathcal{B}}(td) \quad (17.3.2)$$

is the projection of  $td$  on  $\mathcal{B}$  (see Section 12.1.1) and  $d = -D\nabla_x f(x)$  for some appropriate positive definite diagonal scaling matrix  $D$ .

We shall now consider how to find such a point. As we saw in Section 12.1.3, the path  $s(t)$  is a piecewise linear arc, its direction changing every time  $t$  reaches a breakpoint

$$t_i^B = \begin{cases} \min(u_i - x_i, \Delta)/d_i & \text{if } d_i > 0, \\ \max(l_i - x_i, -\Delta)/d_i & \text{if } d_i < 0, \\ 0 & \text{if } d_i = 0 \end{cases}$$

for  $1 \leq i \leq n$ . Suppose that we sort the breakpoints, so that the sorted sequence is  $\{t_0, \dots, t_m\}$ , where  $0 = t_0 < t_1 < \dots < t_m$  and where each  $t_i$  corresponds to one or more of the  $t_j^B$ . The search for the generalized Cauchy point thus corresponds to finding the first interval  $[t_i, t_{i+1})$  containing a local minimizer of  $q(s(t))$ , and this suggests we merely investigate each of these intervals in increasing order.

So, suppose that we have not found such a minimizer for any  $t \in [t_0, t_i]$ . We will now examine interval  $[t_i, t_{i+1})$ . Over this interval, we may express the arc (17.3.2) as

$$s(t) = s^i + \Delta t d^i, \quad (17.3.3)$$

where  $s^i = P_{\mathcal{B}}(t_i d)$ ,

$$d_j^i = \begin{cases} d_j & \text{if } t_j^B > t_i, \\ 0 & \text{otherwise,} \end{cases}$$

and  $\Delta t = t - t_i$ . In addition, (17.3.1) and (17.3.3) show that the value of  $q$  along this portion of the arc is

$$q(s(t)) = q_i + \Delta t \ q'_i + \frac{1}{2}(\Delta t)^2 \ q''_i, \quad (17.3.4)$$

where

$$\begin{aligned} q_i &= f + \langle g, s^i \rangle + \frac{1}{2} \langle s^i, H s^i \rangle, \\ q'_i &= \langle g, d^i \rangle + \langle s^i, H d^i \rangle, \quad \text{and} \\ q''_i &= \langle d^i, H d^i \rangle. \end{aligned} \quad (17.3.5)$$

There are a number of possibilities. Firstly, if

$$\begin{aligned} q'_i &> 0 \quad \text{or} \\ q'_i &= 0 \quad \text{and} \quad q''_i > 0, \end{aligned}$$

the required model minimizer lies at  $t_i$ . If, on the other hand,

$$\begin{aligned} q'_i &\leq 0 \quad \text{and} \quad q''_i \leq 0 \quad \text{or} \\ q'_i &< 0 \quad \text{and} \quad q''_i > 0 \quad \text{and} \quad t_i - q'_i/q''_i \geq t_{i+1}, \end{aligned}$$

the model minimizer lies at or beyond  $t_{i+1}$ . Finally, if

$$q'_i < 0 \quad \text{and} \quad q''_i > 0 \quad \text{and} \quad t_i - q'_i/q''_i < t_{i+1},$$

the required model minimizer is at  $t_i - q'_i/q''_i$ . Thus we can completely classify the interval merely knowing the values of the slope  $q'_i$  and curvature  $q''_i$  of  $q$  along the arc  $s(t)$  at the start of the interval. It remains to show that we may calculate this slope and curvature efficiently. The key is to observe that

$$d^i = d^{i-1} - e^i, \quad \text{where} \quad e^i = \sum_{j \in \mathcal{I}_i} d_j e_j \quad \text{and} \quad \mathcal{I}_i = \{j \mid 1 \leq j \leq n \quad \text{and} \quad t_j^B = t_i\};$$

here  $\mathcal{I}_i$  is the set of indices of components of  $s(t)$  that encounter a bound at  $t = t_i$ . In this case, if we update

$$\langle g, d^i \rangle = \langle g, d^{i-1} \rangle - \langle g, e^i \rangle \quad \text{and} \quad H d^i = H d^{i-1} - H e^i,$$

the remaining constituent parts of  $q'_i$  and curvature  $q''_i$  in (17.3.5) may be obtained by forming the inner products  $\langle s^i, H d^i \rangle$  and  $\langle d^i, H d^i \rangle$ . Thus, by recurring  $\langle g, d^i \rangle$  and  $H d^i$ , we may update the required slope and curvature via two inner products and the matrix-vector product  $H e^i$ . Since  $e^i$  is most likely an extremely sparse vector, the required matrix-vector product can normally be formed most efficiently, especially when  $H$  is sparse or structured.

In summary, we may compute the generalized Cauchy point from the following algorithm, Algorithm 17.3.1.

**Algorithm 17.3.1: Finding the generalized Cauchy point**

**Step 0: Initialization.** The gradient  $g$ , Hessian  $H$ , bounds  $\ell$  and  $u$ , and trust-region radius  $\Delta$  are given, along with the search vector  $d$ . Compute the breakpoints

$$t_i^B = \begin{cases} \min(u_i - x_i, \Delta)/d_i & \text{if } d_i > 0, \\ \max(\ell_i - x_i, -\Delta)/d_i & \text{if } d_i < 0, \\ 0 & \text{if } d_i = 0 \end{cases}$$

for  $i = 1, \dots, n$ . Let  $t_0 = 0$ ,

$$\mathcal{I}_0 = \{j \mid 1 \leq j \leq n \text{ and } t_j^B = 0\} \text{ and } e^0 = \sum_{j \in \mathcal{I}_0} d_j e_j,$$

set  $d^0 = d - e^0$ ,  $s^0 = 0$ , compute  $\langle g, d^0 \rangle$  and  $Hd^0$ , and set  $i = 0$ .

**Step 1: Compute the slope and curvature.** Compute

$$q'_i = \langle g, d^i \rangle + \langle s^i, Hd^i \rangle \text{ and } q''_i = \langle d^i, Hd^i \rangle$$

from the given  $\langle g, d^i \rangle$  and  $Hd^i$ .

**Step 2: Find the next breakpoint.** Determine  $t_{i+1}$ , the first breakpoint beyond  $t_i$ .

**Step 3: Check the current interval for the generalized Cauchy point.**

- If  $q'_i \geq 0$ , set  $s^{GC} = s_i$  and stop.
- If  $q''_i > 0$  and  $\Delta t = -q'_i/q''_i < t_{i+1} - t_i$ , set  $s^{GC} = s_i + \Delta t d_i$  and stop.

**Step 4: Prepare for the next interval.** Let

$$\mathcal{I}_{i+1} = \{j \mid 1 \leq j \leq n \text{ and } t_j^B = t_{i+1}\} \text{ and } e^{i+1} = \sum_{j \in \mathcal{I}_{i+1}} d_j e_j,$$

set  $\Delta t = t_{i+1} - t_i$ ,

$$s^{i+1} = s_i + \Delta t d_i \text{ and } d^{i+1} = d^i - e^{i+1},$$

and update

$$\langle g, d^{i+1} \rangle = \langle g, d^i \rangle - \langle g, e^{i+1} \rangle \text{ and } Hd^{i+1} = Hd^i - He^{i+1}.$$

Increment  $i$  by 1 and go to Step 1.

Notice that the breakpoints do not actually need to be ordered, merely that the next breakpoint should be available. There are many very fast methods for sorting sets of numbers. The heapsort method is particularly appropriate here, as this method

performs a partial sort from which the  $(i+1)$ st smallest member of a set may be found very efficiently once the first  $i$  smallest are known.

### Notes and References for Subsection 17.3.0

This section is based on the algorithm given by Conn, Gould, and Toint (1988b) and forms the basis of the Cauchy point calculation in LANCELOT (see Conn, Gould, and Toint, 1992b, p. 119). Although we have correctly said that the point computed by this algorithm is a suitable generalized Cauchy point, the proof of that statement is not obvious as it involves introducing yet another criticality measure (the projection of the gradient of the subspace defined by the active bounds at  $x_k + s_k^{\text{GC}}$ ). It can be found in Conn, Gould, and Toint (1988a).

Modern sorting schemes require  $O(n \log n)$  operations to sort a set of  $n$  numbers. The heapsort method is due to Williams (1964). For more details on sorting, see the book by Knuth (1973).

## 17.4 Other Numerical Considerations

### 17.4.1 Computing the Model Decrease

A common mistake when using a quadratic model is to compute the difference  $m(x+s) - m(x)$  in the obvious way, that is, by computing both  $m(x+s)$  and  $m(x)$  and then subtracting them. This may be particularly unwise when both values are very large but their difference is small, since numerical rounding can cause significant cancellation errors. If the model is quadratic, it is usually far better in this case to recognize that

$$m(x+s) - m(x) = \langle g, s \rangle + \frac{1}{2} \langle s, Hs \rangle$$

and to compute the difference in the model values directly.

If we are using the (preconditioned) conjugate gradient method, Algorithm 5.1.4, the decrease in the model during iteration  $k$  is

$$\delta m_k = \alpha_k \langle g_k, p_k \rangle + \frac{1}{2} \alpha_k^2 \langle p_k, Hp_k \rangle.$$

It follows from (5.1.56), (5.1.58), and (5.1.59) (p. 88) that

$$\langle g_k, p_k \rangle = \langle v_k, Mp_k \rangle = \langle v_k, M(-v_k + \beta_{k-1} p_{k-1}) \rangle = -\langle v_k, Mv_k \rangle = -\langle v_k, g_k \rangle.$$

Hence (5.1.55) (p. 88) shows that

$$\delta m_k = -\alpha_k \langle g_k, v_k \rangle + \frac{1}{2} \alpha_k^2 \langle p_k, Hp_k \rangle = -\frac{1}{2} \alpha_k \langle g_k, v_k \rangle, \quad (17.4.1)$$

and thus the decrease may be calculated from known quantities at negligible cost.

When the truncated conjugate gradient method, Algorithm 7.5.1, or the generalized Lanczos trust-region method, Algorithm 7.5.2, is used, the decrease (17.4.1) occurs

during iteration  $k$  unless the trust-region boundary is encountered. For this latter special case, the model decrease using the truncated conjugate gradient method is

$$\delta m_k = -\sigma_k \langle g_k, v_k \rangle + \frac{1}{2} \sigma_k^2 \langle p_k, H p_k \rangle,$$

where  $\sigma_k$  is calculated from (7.5.5) (p. 206). Once again, all of the ingredients for determining  $\delta m_k$  are already available, and the decrease may be calculated at negligible additional cost. For the generalized Lanczos trust-region method, the equivalence between the problems (7.5.57) and (7.5.58) (p. 222) shows that the *overall* decrease once the boundary has been reached is

$$\delta m = \gamma_0 \langle h_k, e_1 \rangle + \frac{1}{2} \langle h_k, T_k h_k \rangle.$$

It then follows from (7.5.58) and (7.5.59) that

$$\delta m = \frac{1}{2} \gamma_0 \langle h_k, e_1 \rangle - \frac{1}{2} \lambda_k \|h_k\|_2^2 = \frac{1}{2} \gamma_0 [h_k]_1 - \frac{1}{2} \lambda_k \Delta^2.$$

Since all of the components in this expression are already available, yet again the model decrease is available at almost no extra cost.

There is one final case where the value of the model decrease is essentially available for free, namely, in the computation of the generalized Cauchy point described in Section 17.3. It follows directly from (17.3.4) that the decrease achieved in the interval  $[t_i, t_{i+1})$  is

$$\Delta t q'_i + \frac{1}{2} (\Delta t)^2 q''_i,$$

where  $\Delta t$  is the step taken in the interval and  $q'_i$  and  $q''_i$  are the slope and curvature along the arc computed in Algorithm 17.3.1.

### 17.4.2 Computing the Value of $\rho_k$

Any concerns we have expressed about numerical cancellation errors that might occur when computing the differences in the model apply equally to the difference  $f(x+s) - f(x)$ . In particular, if at all possible, it is best to avoid adding a constant term to the objective function value as this may swamp other, more significant terms.

One of the most dangerous stages for any trust-region method is, perhaps surprisingly, when it is on the point of convergence. In this case both the numerator and denominator in

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{m_k(x_k + s_k) - m_k(x_k)}$$

will be small and most probably will suffer from the effects of floating-point arithmetic. As an extreme example, when the differences are both at the level of machine precision and there is cancellation involved when forming the values the computed values might even have the wrong sign. Thus, rather than having  $\rho_k = 1$ , we might easily see a computed  $\rho_k = -1$  causing our algorithm to reduce the trust-region radius. Unfortunately, the reduction is in vain, as this usually produces steps for which the cancellation is even more pronounced! Thus, we will observe a sequence of unsuccessful iterates, at best terminating when the radius underflows!

In practice, we usually take the precaution of regarding the case when both  $f(x_k + s_k) - f(x_k)$  and  $m_k(x_k + s_k) - m_k(x_k)$  are small as a successful iteration. Thus, for instance, we might replace  $\rho$  by 1 whenever the absolute values of both numerator and denominator are a small multiple of the relative machine precision. Since we normally take precautions to ensure that at least  $m_k(x_k + s_k) - m_k(x_k)$  is computed accurately (see Section 17.4.1), such a strategy is usually effective.

For Newton-like iterations, we normally expect to see  $\rho_k$  approaching 1 in the limit. Occasionally, a value larger than 1 is observed, and this usually indicates that the Hessian is singular at the limit point.

## Notes and References for Subsection 17.4.2

In LANCELOT (see Conn, Gould, and Toint, 1992b), we let  $\delta_k = \epsilon \max(1, |f(x_k)|)$  for some small  $\epsilon > 0$  slightly larger than the relative machine precision  $\epsilon_M$ , and we compute

$$\delta f_k = f(x_k + s_k) - f(x_k) - \delta_k \quad \text{and} \quad \delta m_k = m_k(x_k + s_k) - m_k(x_k) - \delta_k.$$

We then use the value

$$\rho_k = \begin{cases} 1 & \text{if } |\delta f_k| < \epsilon \text{ and } |\delta m_k| < \epsilon \text{ or if } m(x + s) \equiv f(x + s), \\ \frac{\delta f_k}{\delta m_k} & \text{otherwise.} \end{cases}$$

We have found that a value  $\epsilon = 10\epsilon_M$  has been a most effective safeguard against roundoff in practice.

### 17.4.3 Stopping Conditions

#### 17.4.3.1 Unconstrained Problems

Knowing when to stop any of the methods we have considered in this book is of paramount importance. Unfortunately, it is also surprisingly difficult. Both of these comments are especially true when the problem being solved is noisy and/or when function evaluations are very expensive. For the unconstrained minimization of a smooth objective  $f(x)$ , we have shown that the methods we have developed normally have at least one subsequence of gradients that converges to zero. Perhaps, naively, we might then choose to stop the algorithm as soon as

$$\|\nabla_x f(x_k)\| \leq \epsilon_g \tag{17.4.2}$$

for some suitable small positive constant  $\epsilon_g$ . Unfortunately, such a test relies heavily on the fact that both  $f$  and  $x$  are “well scaled”.

To see this, if we multiply  $f$  by a constant  $\sigma$ , the gradient of the scaled problem is  $\sigma \nabla_x f(x_k)$ , and the test (17.4.2) may not be satisfied at  $x_k$ , despite  $x_k$  being just as close to the solution of the scaled problem as to that of the unscaled one. A way around this difficulty is to change the stopping test so that it is a relative instead of absolute test. For example, if  $x^{\text{TYP}}$  is a “typical” value<sup>282</sup> of  $x$ , it may be appropriate

---

<sup>282</sup>Of course, sometimes it is not easy to determine  $x^{\text{TYP}}$ .

to stop when

$$\|\nabla_x f(x_k)\| \leq \epsilon_g \|\nabla_x f(x^{\text{TYP}})\|.$$

Relative stopping rules like this are extremely popular with iterative methods for linear equations.

Bad scaling of the variables may also cause difficulties. For example, if we rescale  $x$  so that  $x_k = S_k \bar{x}_k$  and let  $\bar{f}(\bar{x}_k) = f(S_k \bar{x}_k) = f(x_k)$ , we see that  $\nabla_{\bar{x}} \bar{f}(\bar{x}_k) = S_k \nabla_x f(x_k)$ . Thus, once again, the test (17.4.2) may not be satisfied for  $\bar{x}_k$ , despite the fact that the new variables are simply rescalings of the originals. This lack of scale invariance may be avoided if we change the norm we use to measure the gradient. In particular, the stopping rule

$$\|\nabla_x f(x_k)\|_{(\nabla_{xx} f(x_k))^{-1}} \leq \epsilon_g$$

is often used for small problems, where inverting  $\nabla_{xx} f(x_k)$  is a viable proposition. When there are a large number of unknowns and  $M_k$  is an easily invertible approximation of  $\nabla_{xx} f(x_k)$ , the alternative

$$\|\nabla_x f(x_k)\|_{M_k^{-1}} \leq \epsilon_g$$

is more practical. Notice that measuring such an  $M_k$  norm of the gradient is consistent with rescaling of the trust-region norm (see Sections 6.7 and 8.1.4), and that such norms are frequently used when using iterative methods to find approximate solutions to the trust-region subproblems (see Sections 5.1.6, 7.5.1, and 7.5.4). Since this stopping rule is not invariant to scalings of  $f$ , the relative version

$$\|\nabla_x f(x_k)\|_{M_k^{-1}} \leq \epsilon_g \|\nabla_x f(x^{\text{TYP}})\|_{(M^{\text{TYP}})^{-1}}$$

may be preferred for some “typical” point  $x^{\text{TYP}}$  and matrix  $M^{\text{TYP}}$ .

In practice, we may stop the algorithm for other reasons. As we indicated in Section 17.4.2, unless we are careful, computed values of  $\rho_k$  close to termination may be slightly or catastrophically inaccurate. In the latter case, it can be that the trust-region radius becomes increasingly small without progress being made. This is also possible if the gradient has been incorrectly coded, or if it is subject to noise. An algorithm needs to be able to cope with this possibility, and it is sensible to stop if

$$\Delta_k \leq \epsilon \|x_k\|$$

for some  $\epsilon$  of the order of the relative machine precision,  $\epsilon_M$ . Another case where there is little point in continuing is when the computed step is unlikely to make any difference to the current estimate of the solution. For example, if

$$|[s_k]_i| < \epsilon_M |[x_k]_i|$$

for all  $1 \leq i \leq n$  and  $[x_k]_i \neq 0$ , the floating-point values  $x_k$  and  $x_k + s_k$  are probably indistinguishable, and the iteration should be terminated.

One eventuality that may also arise is that  $f$  may actually be unbounded from below (AF.2 is violated) when the sequence of iterates diverges to infinity. Some authors suggest that bounds which represent infinity should be imposed on every variable, and the problem solved as a bound-constrained minimization. We prefer simply to impose an upper bound on the permissible trust-region radius and to stop if a step to this trust-region boundary proves acceptable. In practice, divergence often (but not always) occurs when negative curvature is discovered, and the divergence is frequently quite rapid. We suggest that if a step along a direction of negative curvature proves to be very successful, internal doubling (see Section 10.5.1) is often worthwhile.

It might also happen that  $f$  is unbounded from below for finite values of the variables. There is little we can do but monitor the values of  $f(x_k)$  as the algorithm proceeds and stop if they become unreasonably small given the problem at hand.

#### 17.4.3.2 Nonlinear Least Squares

When the objective function has the form

$$f(x) = \frac{1}{2} \|c(x)\|_2^2, \quad (17.4.3)$$

the methods we have considered in Section 16.1 normally result in at least one sequence  $\{A^T(x_k)c(x_k)\}$  that converges to zero. In this case, the simplest termination rule would be to stop as soon as

$$\|A^T(x_k)c(x_k)\|_2 \leq \epsilon_g$$

for some suitable small positive constant  $\epsilon_g$ . As before, such a test is not invariant to the scaling of the variables, nor to multiplication of the vector  $c(x)$  by a constant, and a stopping rule like

$$\|A^T(x_k)c(x_k)\|_{(A(x_k)A^T(x_k))^{-1}} \leq \epsilon_g \|A^T(x^{\text{Typ}})c(x^{\text{Typ}})\|_{(A(x^{\text{Typ}})A^T(x^{\text{Typ}}))^{-1}},$$

for some typical value  $x^{\text{Typ}}$ , is more appropriate—notice here that we have replaced the Hessian of  $f$  by its simpler Gauss–Newton approximation. Alternatively, we may prefer a stopping rule like

$$\|A^T(x_k)c(x_k)\|_2 \leq \epsilon_g \|A^T(x_k)\|_2 \max(\|c(x_k)\|_2, \|c(x^{\text{Typ}})\|_2)$$

based on the Cauchy–Schwarz inequality.

We should remind the reader that if our goal is to minimize (17.4.3) we do not allow other scalings of  $c(x)$ , since they implicitly change the problem and hence its solution. However, this once again reinforces our point that if our goal is to find a root of  $c(x)$ , there is nothing sacred about measuring infeasibility in the  $\ell_2$  norm, and a weighted  $\ell_2$  or some other norm is perhaps to be preferred. In addition, as our goal is then to make  $\|c(x)\| = 0$ , we may also choose to stop if

$$\|c(x_k)\| \leq \epsilon_c \|c(x^{\text{Typ}})\|,$$

for another suitable small positive constant  $\epsilon_c$ .

### 17.4.3.3 Constrained Problems

When the problem is constrained, there are three additional considerations. Firstly, and most obviously, we are aiming to satisfy the constraints, and thus we might naively require that

$$\|c(x_k)^{\mathcal{I}^-}\| \leq \epsilon_c \quad (17.4.4)$$

for some suitable small positive constant  $\epsilon_c$ , where  $c(x)^{\mathcal{I}^-}$  is the vector of both equality constraints and inequality constraints<sup>283</sup> whose value is negative at  $x$  (see Sections 11.4 and 14.5). As such a test depends on scalings of the constraints, it is often better to replace (17.4.4) by a relative test like

$$\|c(x_k)^{\mathcal{I}^-}\| \leq \epsilon_c \|c(x^{\text{typ}})\|$$

or preferably a componentwise test of the form

$$|c_i(x_k)^{\mathcal{I}^-}| \leq \epsilon_c |c_i(x^{\text{typ}})|$$

for all  $i$ . Notice that the tests are made relative to  $c(x^{\text{typ}})$  not  $c(x^{\text{typ}})^{\mathcal{I}^-}$ , since we wish to find typical values of the constraint functions, not their infeasibilities.

The second consideration is that the other first-order criticality condition may be written in a number of theoretically equivalent ways. In Section 3.2.2, we saw that a critical point  $x_*$  satisfies

$$\nabla_x f(x_*) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_*]_i \nabla_x c_i(x_*) = 0 \quad (17.4.5)$$

and  $[y_*]_{\mathcal{I}} \geq 0$ , and that we may replace the first of these conditions by the equivalent

$$N_{\mathcal{A}}^T \nabla_x f(x_*) = 0, \quad (17.4.6)$$

where the columns of  $N_{\mathcal{A}}$  are a basis for the null-space of the active constraint gradients. When the constraints describe a convex feasible region  $\mathcal{C}$ , these conditions become

$$P_{\mathcal{C}}[x_* - \nabla_x f(x_*)] - x_* = 0 \quad (17.4.7)$$

so long as  $x_*$  is feasible, where  $P_{\mathcal{C}}$  defines a projection onto  $\mathcal{C}$  (see Section 12.1.4). When the equality and inequality constraints are to be treated differently by the algorithm, we have that

$$N_{\mathcal{E}}^T \left( \nabla_x f(x_*) - \sum_{i \in \mathcal{I}} [y_*]_i \nabla_x c_i(x_*) \right) = 0 \quad (17.4.8)$$

and  $[y_*]_{\mathcal{I}} \geq 0$ , where now  $N_{\mathcal{E}}$  gives a basis for the null-space of the gradients of the equality constraints. When it is just the inequality constraints that define a convex feasible region  $\mathcal{C}$ , we have that

$$P_{\mathcal{C}} \left[ x_* - \nabla_x f(x_*) - \sum_{i \in \mathcal{E}} [y_*]_i \nabla_x c_i(x_*) \right] - x_* = 0 \quad (17.4.9)$$

---

<sup>283</sup>Recall that they are assumed to be of the form  $c_i(x) \leq 0$ .

so long as  $x_* \in \mathcal{C}$ . For many of the algorithms we have considered, we aim to ensure that  $[y_k]_{\mathcal{I}} \geq 0$  throughout, and thus it is simply the remaining equations (17.4.5), (17.4.6), and (17.4.8) that occur in the limit. For the other cases, feasibility of the convex constraints is ensured throughout, and it is just the remaining conditions (17.4.7) and (17.4.9) that happen in the limit. Thus in all cases, writing the relevant criticality condition in the generic form  $\theta(x_*, y_*) = 0$ , we shall choose to stop the relevant algorithm when

$$\|\theta(x_k, y_k)\| \leq \epsilon_g,$$

or better,

$$\|\theta(x_k, y_k)\| \leq \epsilon_g \|\theta(x^{\text{TYP}}, y^{\text{TYP}})\|,$$

or preferably,

$$\|\theta(x_k, y_k)\|_{(M_k)^{-1}} \leq \epsilon_g \|\theta(x^{\text{TYP}}, y^{\text{TYP}})\|_{(M^{\text{TYP}})^{-1}}$$

for some suitable small positive constant  $\epsilon_g$ .

The last consideration is that many of the algorithms we have considered require additional assumptions in order to guarantee that feasibility is actually attained even for a subsequence of the iterates. In practice, it may happen that the problem has no feasible point, at least in the region that the iterates are exploring. If the norm of the infeasibilities appears not to be changing very much, it may be worth monitoring  $\|A^T(x_k)c(x_k)\|$  to see if  $x_k$  is approaching a stationary point of the sum-of-squares of the infeasibilities.

### Notes and References for Subsection 17.4.3

There are good discussions on stopping conditions in the books by Dennis and Schnabel (1983, Section 7.2) and Gill, Murray, and Wright (1981, Section 8.2.3). For nonlinear least squares, see specifically Dennis, Gay, and Welsch (1981) and Dennis and Schnabel (1983, Section 10.4).

### 17.4.4 Noise and/or Expensive Function Evaluations

In some fairly elementary ways, problems that are inherently noisy require special considerations in any implementation. For example, if one has an option for an automatic choice of the initial trust-region radius, in the absence of noise, a value that is inadvertently small for a particular problem merely results in inefficiencies, until a sufficient number of successful steps results in a more reasonable trust-region radius. However, in the presence of noise, this initial choice may lead to complete failure. Indeed, too small a step results in too small a model reduction compared to the noise on the objective function, therefore typically causing the algorithm to stop prematurely. Thus precautions, such as a lower bound related to the noise level on the initial trust-region radius size, are advisable. Similar comments can be made about being careful concerning the choice of the initial Hessian in a trust-region algorithm using a quadratic model, since, for the same reasons, the choice of a large Hessian resulting in a short step may be adequate for a noise-free problem but can be disastrous for a noisy one.

Finally, accurately solving the trust-region subproblem at every iteration is not normally a good idea. However, if the cost of function and gradient evaluation of the underlying problem is significantly higher than the cost of solving this subproblem, then solving it accurately may indeed reduce the number of iterations and is consequently desirable.

## 17.5 Software

We have tried to mention relevant trust-region software packages as we have moved through this book. At this stage, we simply wish to reinforce the impression we hope we have conveyed that there is now a great deal of good optimization software available both as commercial products and in the public domain. An excellent assessment of what was available up until 1993 was given in the book by Moré and Wright (1993), who have subsequently kept their survey up to date online at

<http://www.mcs.anl.gov/otc/Guide/SoftwareGuide/>

Another important and useful innovation is the Network-Enabled Optimization System (NEOS)

<http://www.mcs.anl.gov/otc/Server/>

which provides an Internet-based mechanism for passing optimization problems to and from many of the above-mentioned software packages for solution.

Of course, the theoretical properties of an algorithm amount to very little if the method in question performs poorly in practice. Many of the works cited throughout our book contain the results of tests that purport to show that their associated algorithms are effective. More recently, there have been a number of papers devoted to comparisons between contending software packages, and this itself has led to a better understanding of the roles of the numerous algorithmic choices we have exhibited throughout this book. We warmly welcome this trend and strongly believe that the competitive element this has engendered in many optimization researchers is leading to ever better algorithms and software.

## Notes and References for Section 17.5

The reader should either consult the notes at the ends of individual sections of our book or look at the “software” entry in the index on p. 947 for details of the available trust-region-based software packages of which we are aware. Two additional surveys that have so-far escaped our categorization are that on least-squares methods by Fraley (1989) and that on trust-region methods for constrained optimization by Sadjadi and Ponnambalam (1999).

# Appendix: A Summary of Assumptions

## Unconstrained Minimization

### Assumptions on the Problem

**AF.1** [p. 38]:  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is twice-continuously differentiable on  $\mathbb{R}^n$ .

**AF.1c** [p. 319]:  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is thrice-continuously differentiable on  $\mathbb{R}^n$ .

**AF.2** [p. 121]:  $f(x) \geq \kappa_{\text{lbf}}$  for all  $x \in \mathbb{R}^n$ .

**AF.3** [p. 121]:  $\|\nabla_{xx} f(x)\| \leq \kappa_{\text{ufh}}$  for all  $x \in \mathbb{R}^n$ .

**AF.3b** [p. 272]:  $\omega_k(f, |x_k| s_k) \leq \kappa_{\text{ufh}}$  for all  $k$ .

**AF.4** [p. 319]:  $\|\nabla_{xxx} f(x)\| \leq \kappa_{\text{uft}}$  for all  $x \in \mathbb{R}^n$ .

**AF.5** [p. 333]:  $\|\nabla_x f(x)\| \leq \kappa_{\text{ufg}}$  for all  $x \in \mathbb{R}^n$ .

**AF.6** [p. 277]:  $\nabla_x f$  is uniformly continuous and has values in  $\mathcal{V}$ .

### Assumptions on Norms

**AN.1** [p. 123]:  $\kappa_{\text{une}}^{-1} \|x\| \leq \|x\|_k \leq \kappa_{\text{une}} \|x\|$  for all  $k$  and all  $x \in \mathbb{R}^n$ .

**AN.1b** [p. 252]:  $\|x\|_k \leq \kappa_{\text{une}} \|x\|$  for all  $k$  and all  $x \in \mathbb{R}^n$ .

**AN.1c** [p. 273]:  $\kappa_{\text{cdn}} \pi_k \|x\| \leq \|x\|_k \leq \kappa_{\text{une}} \|x\|$  for all  $k$  and all  $x \in \mathbb{R}^n$ .

**AN.2** [p. 278]:  $\langle y, y \rangle \geq 0$  for all  $y \in \mathcal{V} \cap \mathcal{V}'$ .

**AN.3** [p. 261]:  $|\|g_k\|_{[k]} - \|g_t\|_{[t]}| \rightarrow 0$  whenever  $\|x_k - x_t\| \rightarrow 0$ .

## Assumptions on the Model

**AM.1** [p. 122]: For all  $k$ ,  $m_k$  is twice differentiable on  $\mathcal{B}_k$ .

**AM.1b** [p. 308]: For all  $k$ ,  $m_k$  is twice differentiable on  $\mathbb{R}^n$ .

**AM.1c** [p. 319]: For all  $k$ ,  $m_k$  is thrice differentiable on  $\mathbb{R}^n$ .

**AM.2** [p. 122]:  $m_k(x_k) = f(x_k)$  for all  $k$ .

**AM.3** [p. 122]:  $g_k = \nabla_x f(x_k)$  for all  $k$ .

**AM.3b** [p. 280]:  $\|\nabla_x f(x_k) - g_k\| \leq \kappa_{\text{egg}} \|g_k\|$  for all  $k$ .

**AM.3c** [p. 292]:  $\|\nabla_x f(x_k) - g_k\| \leq \kappa_{\text{egg}} \min[\|g_k\|, \|\nabla_x f(x_k)\|]$  for all  $k$ .

**AM.4** [p. 122]:  $\|\nabla_{xx} m_k(x)\| \leq \kappa_{\text{umh}} - 1$  for all  $x \in \mathcal{B}_k$  and all  $k$ .

**AM.4b** [p. 252]:  $|\langle w, \nabla_{xx} m_k(x)w \rangle| \leq \kappa_{\text{umh}} - 1 \|w\|_k^2$  for all  $x \in \mathcal{B}_k$ , all  $k$  and all  $w \neq 0$ .

**AM.4c** [p. 272]:  $\max[\omega_k(m_k, x_k, g_k), \omega_k(m_k, x_k, s_k), \omega_k(m_k, x_k, u_k)] \leq \kappa_{\text{umh}} - 1$  for all  $k$ .

**AM.4d** [p. 283]:  $\sum_{k=0}^{\infty} \frac{1}{\theta_k} = \infty$ .

**AM.4f** [p. 290]:  $\lim_{\substack{k \rightarrow \infty \\ k \in \mathcal{S}}} \varphi_k[f(x_k) - f(x_{k+1})] = 0$ .

**AM.4g** [p. 308]:  $\|\nabla_{xx} m_k(x)\| \leq \kappa_{\text{umh}} - 1$  for all  $x \in \mathbb{R}^n$  and all  $k$ .

**AM.5** [p. 143]:  $\lim_{k \rightarrow \infty} \|\nabla_{xx} f(x) - \nabla_{xx} m_k(x)\| = 0$  when  $\lim_{k \rightarrow \infty} \|g_k\| = 0$ .

**AM.5b** [p. 255]:  $\lim_{k \rightarrow \infty} \sup_{w \neq 0} \frac{|\langle w, [\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)]w \rangle|}{\|w\|_k^2} = 0$  if  $\lim_{k \rightarrow \infty} \pi_k = 0$ .

**AM.6** [p. 153]:  $\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\| \leq \kappa_{\text{lch}} \|x - y\|$  for  $x, y \in \mathcal{B}_k$ .

**AM.6b** [p. 255]:  $\sup_{w \neq 0} \frac{|\langle w, [\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)]w \rangle|}{\|w\|_k^2} \leq \kappa_{\text{lch}} \|x - y\|_k$  for  $x, y \in \mathcal{B}_k$ .

**AM.7** [p. 309]: The validity of  $m_k$  in  $\mathcal{Q}_k(\delta)$  may be checked for each  $k$  and any  $\delta > 0$ .

**AM.8** [p. 309]: The model is valid after a finite number of improvement steps.

**AM.9** [p. 319]:  $\|\nabla_{xxx} m_k(x)\| \leq \kappa_{\text{umt}}$  for all  $x \in \mathbb{R}^n$  and all  $k$ .

**AM.10** [p. 337]:  $\alpha_0 = 0$  and  $\alpha_k \leq \min[\bar{\alpha}, \kappa_{\alpha} \|s_{k-1}\|^{\psi}]$  for all  $k \geq 1$ .

**AM.11** [p. 337]:  $q_k$  is entirely determined by  $x_k$  and  $x$ . Furthermore, for all  $k$  and all  $x \in \mathcal{B}_k$ ,  $q_k(x_k) = f(x_k)$ ,  $\nabla_x f(x_k) = \nabla_x q_k(x_k)$ , and  $\|\nabla_{xx} q_k(x)\| \leq \kappa_{\text{umh}}$ .

## Assumptions on the Algorithm

**AA.1** [p. 131]: For all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \|g_k\| \min \left[ \frac{\|g_k\|}{\beta_k}, \Delta_k \right].$$

**AA.1b** [p. 251]: For all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{mdc}} \pi_k \min \left[ \frac{\pi_k}{\beta_k^\pi}, \Delta_k \right].$$

**AA.2** [p. 153]: If  $\tau_k = \lambda_{\min}[\nabla_{xx} m_k(x_k)] < 0$ , then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} | -\tau_k | \min[\tau_k^2, \Delta_k^2].$$

**AA.2b** [p. 255]: If  $\tau_k = \inf_{w \neq 0} \frac{\langle w, \nabla_{xx} m_k(x_k) w \rangle}{\|w\|_k^2} < 0$ , then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} | -\tau_k | \min[\tau_k^2, \Delta_k^2, \kappa_{\text{lsd}}^2].$$

**AA.3** [p. 158]: If  $\rho_k \geq \eta_2$  and  $\Delta_k \leq \Delta_{\max}$ , then  $\Delta_{k+1} \in [\gamma_3 \Delta_k, \gamma_4 \Delta_k]$  ( $\gamma_4 \geq \gamma_3 > 1$ ,  $\Delta_{\max} > 0$ ).

**AA.4** [p. 284]: If  $\rho_k \geq \eta_2$ , then  $\Delta_{k+1} \leq \gamma_5 \Delta_k$  ( $\gamma_5 \geq 1$ ).

## Assumptions on the Iterates

**AI.1** [p. 155]: The sequence of iterates  $\{x_k\}$  lies within a closed, bounded domain  $\Omega$ .

## Systems of Nonlinear Equations

### Assumptions on the Problem

**AC.1e** [p. 750]:  $c(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is twice-continuously differentiable on  $\mathbb{R}^n$ .

**AC.3e** [p. 750]:  $\|c(x)\| \leq \kappa_{\text{udec}}$ ,  $\|\nabla_x c(x)\| \leq \kappa_{\text{udec}}$ ,  $\|\nabla_{xx} c(x)\| \leq \kappa_{\text{udec}}$  ( $x \in \mathbb{R}^n$ ).

**AC.5e** [p. 774]:  $\|\nabla_x h(x)\| \leq \kappa_{\text{cmp}}$  and  $\|\nabla_x g(x)\| \leq \kappa_{\text{cmp}}$  ( $x \in \mathbb{R}^n$ ).

### Assumptions on Optimality

**AO.1e** [p. 764]: The vectors  $\{A^T(x_*)(p_i - p_j)\}_{i \in \mathcal{A}(x_*) \setminus \{j\}}$  are linearly independent, where  $j$  is any member of  $\mathcal{A}(x_*)$ .

**AO.3e** [p. 762]: The relationships  $A^T(x_*)y_* = 0$  and  $\langle s, \sum_{i=1}^n [y_*]_i \nabla_{xx} c_i(x_*) s \rangle > 0$  hold for all  $s$  in

$$\mathcal{D} = \left\{ d \mid \max_{y \in \partial h(c(x_*))} \langle d, (\nabla_x c(x_*))^T y \rangle = 0 \text{ and } \|d\| = 1 \right\}$$

at  $x_* \in \mathcal{X}$  and  $y_* \in \partial h(c(x_*))$ .

**AO.4e** [p. 764]: The generalized gradient  $y_*$  lies in the interior of  $\partial h(c(x_*))$ .

## Assumptions on the Model

**AM.1e** [p. 752]: For all  $k$ ,  $m_k^c$  is twice-continuously differentiable on  $\mathcal{B}_k$ .

**AM.2e** [p. 752]:  $m_k^c(x_k) = c(x_k)$  for all  $k$ .

**AM.3e** [p. 752]:  $A_k = A(x_k)$  for all  $k$ .

**AM.4e** [p. 753]:  $\|\nabla_{xx}m_{ik}^c(x)\| \leq \kappa_{\text{umh}} - 1$  for all  $k$ ,  $x \in \mathcal{B}_k$ .

**AM.5e** [p. 753]:  $\lim_{k \rightarrow \infty} \|\nabla_{xx}c_i(x_k) - \nabla_{xx}m_{ik}^c(x_k)\| = 0$  if  $\lim_{k \rightarrow \infty} \|A_k^T m_{ik}^c(x_k)\| = 0$ .

## Assumptions on the Iterates

**AI.2e** [p. 762]: The sequence  $\{x_k\}$  has a single limit point  $x_*$ .

## Assumptions on the Algorithm

**AA.1e** [p. 762]: The step  $s$  is chosen as a local minimizer of the model for which AA.1n holds.

**AA.1f** [p. 765]: The step  $s_k$  is ultimately chosen as the locally unique minimizer of the model for all  $x_k - x_*$  sufficiently small.

**AA.11e** [p. 764]: The sequence  $\{\Delta_k\}$  is such that  $\Delta = \liminf_{k \rightarrow \infty} \Delta_k > 0$ .

## Partially Separable Problems

### Assumptions on the Problem

**AF.1s** [p. 370]:  $f : \mathbb{R}^n \rightarrow \mathbf{R}$  is partially separable and each  $f_i$  is twice-continuously differentiable on  $\mathbb{R}^n$  ( $i \in \{1, \dots, p\}$ ).

**AF.3s** [p. 370]:  $\|\nabla_{xx}f_i(x)\| \leq \kappa_{\text{ufh}}$  for all  $k$ , all  $x \in \mathcal{R}_i$  and all  $i \in \{1, \dots, p\}$ .

**AF.5s** [p. 370]:  $\|\nabla_x f_i(x)\| \leq \kappa_{\text{ufh}}$  for all  $k$ , all  $x \in \mathcal{R}_i$  and all  $i \in \{1, \dots, p\}$ .

## Assumptions on the Model

**AM.1s** [p. 370]: For all  $k$ ,  $m_k$  is partially separable and  $m_{i,k}$  is twice-continuously differentiable on  $\mathcal{B}_k$  ( $i \in \{1, \dots, p\}$ ).

**AM.2s** [p. 370]:  $m_{i,k}(x_k) = f_i(x_k)$  for all  $k$  and all  $i \in \{1, \dots, p\}$ .

**AM.3s** [p. 370]:  $g_{i,k} = \nabla_x f_i(x_k)$  for all  $k$  and all  $i \in \{1, \dots, p\}$ .

**AM.4s** [p. 370]:  $\|\nabla_{xx}m_{i,k}(x)\| \leq \kappa_{\text{umh}} - 1$  for all  $x \in \mathcal{B}_k$ , all  $k$  and all  $i \in \{1, \dots, p\}$ .

## Assumptions on the Algorithm

**AA.1s** [p. 363]: For all  $k$ ,

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{smd}} \|g_k\| \min \left\{ \frac{\|g_k\|}{\beta_k}, \max \left[ \Delta_k^{\min}, \frac{\|s_k\|}{\nu_k^S} \right] \right\}.$$

**AA.2s** [p. 375]: If  $\tau_k = \lambda_{\min}[\nabla_{xx}m_k(x_k)] < 0$ , then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min[\tau_k^2, \nu_k^E \Delta_k^{\min}]^2.$$

## Nonsmooth Problems

### Assumptions on the Problem

**AF.1n** [p. 407]:  $f(x)$  is locally Lipschitz continuous and regular on  $\mathbb{R}^n$ .

### Assumptions on Optimality

**AO.3n** [p. 48]: The conditions  $\nabla_x f(x_*) + (\nabla_x c(x_*))^T y_* = 0$  and

$$\langle d, (\nabla_{xx}f(x_*) + \sum_{i=1}^n [y_*]_i \nabla_{xx}c_i(x_*)) d \rangle > 0 \text{ hold for all } d \text{ in}$$

$$\mathcal{D} = \left\{ d \mid \max_{y \in \partial h(c(x_*))} \langle d, \nabla_x f(x_*) + (\nabla_x c(x_*))^T y \rangle = 0 \text{ and } \|d\| = 1 \right\}$$

at  $x_* \in \mathcal{X}$  and  $y_* \in \partial h(c(x_*))$ .

### Assumptions on the Model

**AM.1n** [p. 408]: The model  $m(x, p, s)$  is locally Lipschitz continuous and regular with respect to  $s$  for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ , and continuous in  $(x, p)$  for all  $s \in \mathbb{R}^n$ .

**AM.2n** [p. 409]:  $m(x, p, 0) = f(x)$  for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ .

**AM.3n** [p. 409]:  $m_d^o(x, p, 0) = f_d^o(x)$  for all  $d \neq 0 \in \mathbb{R}^n$  for all  $(x, p) \in \mathbb{R}^n \times \mathcal{P}$ .

**AM.4n** [p. 409]: The set of parameters  $\mathcal{P}$  is closed and bounded.

### Assumptions on the Algorithm

**AA.1n** [p. 413]: For any given  $x_*$ , there exist constants  $\delta, \epsilon > 0$  and  $\kappa_{\text{mdc}} \in (0, 1)$ , such that the step  $s_k$  satisfies

$$m(x_k, p_k, 0) - m(x_k, p_k, s_k) \geq \kappa_{\text{mdc}} \|g(x_k)\| \min [\delta, \Delta_k],$$

whenever  $x_k \in \mathcal{O}_\epsilon(x_*)$ .

## Constrained Problems

### Assumptions on the Problem

**AC.1** [p. 41]:  $c_i(x)$  are twice-continuously differentiable on  $\mathbb{R}^n$  for all  $i \in \mathcal{E} \cup \mathcal{I}$ .

**AC.1c** [p. 493]: For all  $\mu > 0$ ,  $b(x, \mu)$  is twice-continuously differentiable on  $\text{ri}\{\mathcal{C}\}$ .

**AC.2** [p. 30]:  $\mathcal{C}$  is nonempty, closed, and convex.

**AC.2b** [p. 492]:  $\text{ri}\{\mathcal{C}\} \neq \emptyset$ .

**AC.2c** [p. 536]:  $\text{si}\{\mathcal{C}\} \neq \emptyset$ .

**AC.4** [p. 482]: For all  $x \in \mathcal{C}$  and  $i \in \{1, \dots, m\}$ ,  $\|\nabla_{xx} c_i(x)\| \leq \kappa_{\text{uch}}$ .

**AC.4c** [p. 494]:  $\|\nabla_{xx} b(x, \mu)\| \leq \kappa_{\text{bbh}}(\epsilon, \mu)$  for all  $x \in \mathcal{C} \mid \text{dist}(x, \partial\mathcal{C}) \geq \epsilon$  and all  $\mu, \epsilon > 0$ .

**AC.5** [p. 536]: For all  $x \in \mathcal{C}$  and  $i \in \{1, \dots, m\}$ ,  $\|\nabla_x c_i(x)\| \leq \kappa_{\text{ubgc}}$ .

**AC.6** [p. 482]: For all  $x, v \in \mathcal{C}$  and  $i \in \{1, \dots, m\}$ ,  $\|\nabla_{xx} c_i(x) - \nabla_{xx} c_i(v)\| \leq \kappa_{\text{lcc}} \|x - v\|$ .

**AC.6b** [p. 503]: For all  $x, y \in \mathcal{B}_{k,j}$ ,  $\|\nabla_{xx} m_{k,j}^b(x) - \nabla_{xx} m_{k,j}^b(y)\| \leq \kappa_{\text{lch}} \|x - y\|$ .

**AC.7** [p. 46]:  $\mathcal{C} = \bigcap_{i=1}^m \mathcal{C}_i$ , where  $\mathcal{C}_i = \{x \in \mathbb{R}^n \mid c_i(x) \geq 0\}$ , where each  $c_i : \mathbb{R}^n \rightarrow \mathbb{R}$ .

**AC.8** [p. 494]:  $\lim_{p \rightarrow \infty} b(y_p, \mu) = +\infty$  for any  $\mu > 0$  and any sequence  $\{y_p\}_{p=0}^\infty$  such that  $\lim_{k \rightarrow \infty} \text{dist}(y_p, \partial\mathcal{C}) = 0$ .

### Assumptions on Optimality

**AO.1** [p. 40]: A first-order constraint qualification holds at  $x_*$ .

**AO.1b** [p. 42]: The Jacobian of active constraint gradients,  $\nabla_x c_{\mathcal{A}(x_*)}(x_*)$ , is of full rank.

**AO.1c** [p. 460]: For all  $x_* \in \mathcal{L}_*$ , the vectors  $\{\nabla_x c_i(x_*)\}_{i \in \mathcal{A}(x_*)}$  are linearly independent.

**AO.2** [p. 40]: A second-order constraint qualification holds at  $x_*$ .

**AO.3** [p. 42]: The conditions

$$\begin{aligned} \nabla_x f(x_*) &= \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_*]_i \nabla_x c_i(x_*), \\ c_i(x_*) &= 0 \text{ for all } i \in \mathcal{E}, \\ c_i(x_*) &\geq 0 \text{ and } [y_*]_i \geq 0 \text{ for all } i \in \mathcal{I}, \\ c_i(x_*)[y_*]_i &= 0 \text{ for all } i \in \mathcal{I}, \\ \text{and } \langle s, \nabla_{xx} \ell(x_*, y_*)s \rangle &> 0 \text{ for all } s \in \mathcal{N}_+ \ (s \neq 0) \end{aligned}$$

hold at  $(x_*, y_*)$ , where  $\mathcal{N}_+$  is

$$\left\{ s \in \mathbb{R}^n \mid \begin{array}{l} \langle s, \nabla_x c_i(x_*) \rangle = 0 \text{ for all } i \in \mathcal{E} \cup \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j > 0\} \\ \text{and } \langle s, \nabla_x c_i(x_*) \rangle \geq 0 \text{ for all } i \in \{j \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_j = 0\} \end{array} \right\}.$$

**AO.3b** [p. 578]: The augmented matrix  $K(x_*, y^{\text{LS}}(x_*), 0)$  is nonsingular for all  $x_* \in \mathcal{L}_*$ .

**AO.3c** [p. 591]: The augmented matrix  $K(x_*, y^{\text{LS}}(x_*), 0)$  has inertia  $(n, m, 0)$  for all  $x_* \in \mathcal{L}_*$ .

**AO.4** [p. 42]:  $\{i \in \mathcal{A}(x_*) \cap \mathcal{I} \mid [y_*]_i = 0\} = \emptyset$ .

**AO.4b** [p. 460]: For all  $x_* \in \mathcal{L}_*$ ,  $-\nabla_x f(x_*) \in \text{ri}\{\mathcal{N}(x_*)\}$ .

## Assumptions on Merit Functions

**AW.1** [p. 575]: The functions  $f(x)$  and  $c_i(x)$  are twice-continuously differentiable functions of  $x$ .

**AW.1b** [p. 578]: The second derivatives of the functions  $f(x)$  and  $c_i(x)$  exist and are Lipschitz continuous.

**AW.1c** [p. 668]: The first derivatives of the functions  $f(x)$  and  $c_i(x)$  exist and are Lipschitz continuous in some open set  $\Omega$  containing the iterates generated by the algorithm.

**AW.1d** [p. 670]: The function  $f(x)$  is uniformly bounded from below, the functions  $g(x)$  and  $c(x)$  are uniformly bounded, and  $c(x)$  is Lipschitz continuous, at all points computed by the algorithm.

**AW.1f** [p. 699]: The function  $f(x)$  is uniformly bounded from below, while the functions  $g(x)$ ,  $c(x)$ , and  $A(x)$  are uniformly bounded, at all points computed by the algorithm.

**AW.1g** [p. 742]: The functions  $f$  and  $c$  are twice-continuously differentiable on an open set containing  $\mathcal{C}(\vartheta^{\max})$ , (15.5.44) holds, and their first and second derivatives are uniformly bounded over  $\mathcal{C}(\vartheta^{\max})$ .

**AW.2** [p. 503]:  $\phi(x, \mu_k)$  is bounded below on  $\mathcal{C}$ ; that is, there exists a constant  $\kappa_{\text{lbb}}$  such that  $\phi(x, \mu_k) \geq \kappa_{\text{lbb}}$  for all  $k$  and all  $x \in \mathcal{C}$ .

**AW.3** [p. 607]:  $\nabla_{xx} \Phi(x_*, y_*, \mu_*)$  is positive definite.

**AW.4** [p. 638]: The penalty parameter satisfies  $\sigma > \min_{y \in \mathcal{Y}(x_*)} \|y\|_{\mathbb{D}}$ , where  $\mathcal{Y}(x_*)$  is the set of all Lagrange multipliers at  $x_*$ .

## Assumptions on the Model

**AM.4h** [p. 498]:  $\|\nabla_{xx}m_{k,j}^b(x, \mu_k)\| \leq \kappa_{\text{bbmh}}(\epsilon, \mu_k)$  for all  $x \in \mathcal{B}_{k,j} \cap \mathcal{C} | \text{dist}(x, \partial\mathcal{C}) \geq \epsilon$  and all  $\epsilon > 0$ .

**AM.4i** [p. 499]:  $\|\nabla_{xx}m_{k,j}^f(x)\| \leq \kappa_{\text{umh}} - 1$  for all  $x \in \mathcal{B}_{k,j} \cap \text{ri}\{\mathcal{C}\}$ .

**AM.4j** [p. 638]: The sequence of second derivative approximations  $\{H_k\}$  is bounded.

**AM.5c** [p. 503]: For all  $k$ , if  $\lim_{j \rightarrow \infty} \|\nabla_x m_{k,j}^f(x_{k,j}) + \nabla_x m_{k,j}^b(x_{k,j}, \mu_k)\| = 0$ , then

$$\lim_{j \rightarrow \infty} \|\nabla_{xx}f(x_{k,j}) - \nabla_{xx}m_{k,j}^f(x_{k,j})\| = \lim_{j \rightarrow \infty} \|\nabla_{xx}b(x_{k,j}, \mu_k) - \nabla_{xx}m_{k,j}^b(x_{k,j})\| = 0.$$

## Assumptions on the Algorithm

**AA.1c** [p. 640]: The step  $s$  is chosen as a local minimizer of the model for which AA.1n holds.

**AA.1d** [p. 641]: The step  $s_k$  is ultimately chosen as the locally unique minimizer of the model for all  $x_k - x_*$  sufficiently small.

**AA.1g** [p. 662]: For all  $k$ ,  $A(x_k)n_k^C + c(x_k) = 0$  and  $\|n_k^C\|_2 \leq \kappa_{\text{bsc}}\|c(x_k)\|_2$ .

**AA.1h** [p. 662]: Given  $\xi^N, \theta \in (0, 1]$ ,

$$\alpha_k \in \left[ \min \left[ 1, \frac{\theta \xi^N \Delta_k}{\|n_k^C\|_2} \right], \min \left[ 1, \frac{\xi^N \Delta_k}{\|n_k^C\|_2} \right] \right].$$

**AA.1i** [p. 665]: For all  $k$ ,

$$\delta m_k^T \geq \kappa_{\text{tmd}} \|g^N(x_k)\| \min \left[ \frac{\xi^T \Delta_k}{\|N(x_k)\|}, \frac{\|g^N(x_k)\|}{\beta_k^T} \right].$$

**AA.1j** [p. 696]: For all  $k$ ,

$$\delta m_k^N \geq \kappa_{\text{nmd}} \frac{\|A^T(x_k)c(x_k)\|_2}{\|c(x_k)\|_2} \min \left[ \xi^N \Delta_k, \frac{\|A^T(x_k)c(x_k)\|_2}{\beta_k^N} \right].$$

**AA.1k** [p. 697]: For all  $k$ ,  $\|n_k^N\| \leq \kappa_{\text{drs}}\|n_k^R\|$  and  $n_k^R = 0$  whenever  $A^T(x_k)c(x_k) = 0$ , where  $n_k = R_k n_k^R + N_k n_k^N$ , and where  $R_k$  and  $N_k$  are bases for the null- and range-spaces of  $A(x_k)$  with uniformly bounded nonzero singular values.

**AA.1l** [p. 722]: If  $\{x_{k_i}\}$  is any subsequence of iterates for which  $\lim_{i \rightarrow \infty} \vartheta_{k_i} = 0$ , then  $n_{k_i}$  exists for  $i$  sufficiently large, and  $\|n_{k_i}\| \leq \kappa_{\text{usc}}\vartheta_{k_i}$ .

**AA.1m** [p. 724]: For all  $k$ ,

$$m_k(x_k^N) - m_k(x_k^N + t_k) \geq \kappa_{\text{tmd}} \chi_k \min \left[ \frac{\chi_k}{\beta_k}, \Delta_k \right].$$

**AA.2c** [p. 487]: If  $\tau_k = \lambda_{\min}[Z(x_k)^T \nabla_{xx} \ell(x_k, y_k) Z(x_k)] < 0$ , then

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\text{sod}} |\tau_k| \min[1, \tau_k^2, \sigma_k^2, \Delta_k^2]$$

when  $\|y_k - y^{\text{LS}}\|$  and  $\|P_{\mathcal{T}(x_k)}[-g_k]\|$  are sufficiently small compared to  $|\tau_k|$ .

**AA.2d** [p. 691]: If  $\tau_k = \lambda_{\min}[H^N(x_k)] < 0$ , then

$$m^T(x_k, H_k, 0) - m^T(x_k, H_k, t_k) \geq -\kappa_{\text{sod}} \tau_k \Delta_k^2.$$

**AA.5** [p. 461]:  $\mathcal{A}(x_k^{\text{GC}}) \subseteq \mathcal{A}(x_k + s_k)$ .

**AA.6** [p. 481]:  $y_k$  is consistent for  $k$  large, and  $\lim_{k \rightarrow \infty} \|y_k - y_k^{\text{LS}}\| = 0$  when  $\lim_{k \rightarrow \infty} \pi_k = 0$ .

**AA.7** [p. 522]: For each  $k \geq 0$ ,  $[z_{k,j}]_i \leq \kappa_{\text{uzi}} \max \left[ \frac{1}{[x_{k,j}]_i}, 1 \right]$  for all  $j \geq 0$  and all  $i$ .

**AA.8** [p. 522]: For each  $k \geq 0$ ,  $\lim_{j \rightarrow \infty} \|z_{k,j} - \mu_k X_{k,j}^{-1} e\| = 0$  if  $\lim_{j \rightarrow \infty} \|g_{k,j}\| = 0$ .

**AA.9** [p. 596]: The sequence  $\{\eta_i\}$  satisfies the bound

$$\sum_{l=1}^{k_{i+1}-k_i} \eta_{k_i+l} \leq \kappa_\eta \mu_{k_i+1}^{\alpha_\eta}$$

for some positive  $\kappa_\eta$  and  $\alpha_\eta$  and all  $i \geq 0$ , where  $\mathcal{K} = \{k_0, k_1, k_2, \dots\}$  is the set of the indices of the iterations at which Step 3 of Algorithm 14.4.1 is executed.

**AA.10** [p. 599]: The penalty parameters satisfy  $\mu_{k+1} = \mu_k$  for all  $k \geq k_1$  and some  $k_1 \geq 0$ .

**AA.11** [p. 640]: The sequence  $\{\Delta_k\}$  is such that  $\Delta = \liminf_{k \rightarrow \infty} \Delta_k > 0$ .

**AA.12** [p. 666]: There is a constant  $\kappa_{\text{bns}} > 0$  such that for all  $k$ ,

$$\kappa_{\text{bns}}^{-1} \leq \sigma_{\min}[N(x_k)] \leq \sigma_{\max}[N(x_k)] \leq \kappa_{\text{bns}}.$$

**AA.13** [p. 669]: There is a constant  $\sigma_{\max}^B > 0$  such that for all  $k$ ,

$$0 \leq \sigma_k^B \leq \sigma_{\max}^B.$$

**AA.14** [p. 702]: There is a constant  $\kappa_{\text{bns}} > 1$  such that for all  $k$ ,

$$\kappa_{\text{bns}}^{-1} \leq \sigma_{\min}[A(x_k)] \leq \sigma_{\max}[A(x_k)] \leq \kappa_{\text{bns}}.$$

## Assumptions on the Iterates

**AI.2** [p. 638]: The sequence  $\{x_k\}$  has a limit point  $x_*$  for which  $c(x_*)^{\mathcal{I}^-} = 0$ .

**AI.2b** [p. 640]: The sequence  $\{x_k\}$  has a single limit point  $x_*$  for which  $c(x_*)^{\mathcal{I}^-} = 0$ .

# Afterword

We have now completed our tour of trust-region methods, building up some familiarity with the concept and its applications to various interesting optimization problems. Realistically, however, we freely admit that we have not investigated all possible avenues and ideas. Moreover, we have not done full justice to many of the interesting proposals mentioned in the notes at the end of each section. We only hope that our book has conveyed some of our enthusiasm and will foster interest in the subject, despite the gaps that we have undoubtedly left. The field is ripe for further explorations and much remains to be done. We conclude, this time for good, by urging the reader to take this book for what it really is: our manner of paving the way, as best as we can, to better concepts, better algorithms, and, ultimately, better understanding of methods for nonlinear optimization.

# Annotated Bibliography

The following works have been cited in our book. Some are general references to background material, while others are central to the development of the trust-region methods we have covered. For those references directly relating to trust-region methods, we have included a short summary of the work's contents. We have deliberately not included any but the most relevant of the literally thousands of citations to the Levenberg (1944)–Morrison (1960)–Marquardt (1963) method. The complete L<sup>A</sup>T<sub>E</sub>X bibliography, together with up-to-date additions, is available online at

`ftp://ftp.numerical.rl.ac.uk/pub/trbook/trbook.bib`

and

`ftp://thales.math.fundp.ac.be/pub/trbook/trbook.bib`

We would be delighted to receive any corrections or updates to this list.

N. M. Alexandrov (1998). A trust-region algorithm for bilevel optimization. Presentation at the Optimization '98 Conference, Coimbra, Portugal.

**Summary.** A trust-region method for solving nonlinear bilevel optimization problems, without any special assumptions on structure, such as separability or convexity, is presented. The algorithm is related to descent methods on the upper level problem which use some gradient information from the lower level problem and can be extended to multilevel optimization. It is motivated by applications in multidisciplinary optimization. The algorithm, its analysis, and a number of numerical examples are discussed and contrasted with the collaborative optimization method, in which the bilevel program is converted into a single-level problem by using the Karush–Kuhn–Tucker conditions of the lower level system as constraints for the upper level problem.

N. M. Alexandrov and J. E. Dennis (1994a). Algorithms for bilevel optimization. Technical Report 94–77, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA.

**Summary.** Two approaches for solving general bilevel optimization problems are presented. Both use a trust-region globalization strategy, and can be extended to handle the general multilevel problem. No convexity assumptions are made, but it is assumed that the problem has a nondegenerate feasible set. Necessary optimality conditions for the bilevel problem formulations are considered and the results extended to obtain multilevel optimization formulations with constraints at each level.

N. M. Alexandrov and J. E. Dennis (1994b). Multilevel algorithms for nonlinear optimization. Technical Report 94-53, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center Hampton, VA, USA.

**Summary.** Multidisciplinary design optimization (MDO) gives rise to nonlinear optimization problems with a large number of constraints that occur in blocks. A class of multilevel optimization methods, motivated by their structure and number and by the expense of the derivative computations for MDO, is proposed. The algorithms are an extension of the local Brown (1969)–Brent (1973) algorithms for nonlinear equations. They allow the user to partition constraints into arbitrary blocks to fit the application, and they separately process each block and the objective function, restricted to certain subspaces. The methods use a trust-region globalization strategy and are globally convergent.

- N. M. Alexandrov and J. E. Dennis (1999). A class of general trust-region multilevel algorithms for nonlinear constrained optimization: Global convergence analysis. Technical Report TR99786-S, Center for Research on Parallel Computers, Houston, TX, USA.

**Summary.** A globally convergent class of trust-region multilevel algorithms for solving equality constrained problems is presented, motivated by engineering optimization problems with dense subproblem structure. The constraints are partitioned into blocks. At every iteration, a multilevel algorithm minimizes models of each of the constraint blocks, followed by a model of the objective function within these blocks, each yielding a substep. The trial step is the sum of these substeps. Each substep is required to satisfy sufficient decrease and boundedness conditions on its restricted model and can be computed by a specialized method. The trial step is evaluated via one of two merit functions that take into account the autonomy of subproblem processing.

- N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon (1998). A trust region framework for managing the use of approximation models. *Structural Optimization*, **15**(1), 16–23.

**Summary.** A robust, globally convergent approach to using approximation models of various fidelities is presented. It is based on trust regions and converges to a solution of the original high-fidelity problem. The method suggests ways to decide when the fidelity, and thus the cost, of the approximations might be altered in the course of the iterations. No assumptions on the structure of the original problem, such as convexity or separability, are made, and the requirements on the approximations are mild. These can be of any nature appropriate to an application; for instance, they can be represented by analyses, simulations, or simple algebraic models.

- F. Alizadeh (1995). Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, **5**(1), 13–51, 1995.

- D. M. Allen (1995). Tailoring nonlinear least squares algorithms for the analysis of compartment models. In J. Sall and A. Lehman, eds., *Computationally Intensive Statistical Methods. Proceedings of the 26th Symposium on the Interface*, Fairfax Station, VA, USA, Vol. 26, The Interface Foundation of North America, Fairfax, VA, pp. 533–535.

**Summary.** Compartment models are defined by linear differential equations and are widely used in pharmacokinetics. It is shown how general optimization methods, including linesearch and trust-region methods, can be tailored to exploit the special structure of such models.

- E. L. Allgower, K. Böhmer, F. A. Potra, and W. C. Rheinboldt (1986). A mesh-independence principle for operator equations and their discretizations. *SIAM Journal on Numerical Analysis*, **23**(1), 160–169.

- J. Amaya (1985). On the convergence of curvilinear search algorithms in unconstrained optimization. *Operations Research Letters*, **4**(1), 31–34.

**Summary.** The conditions under which curvilinear algorithms for unconstrained optimization converge are unified. Two gradient path approximation algorithms and a trust-region curvilinear algorithm are examined in this light.

- E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu (1996). Implementation of interior point methods for large scale linear programming. In T. Terlaky, ed., *Interior Point Methods in Mathematical Programming*, pp. 189–252, Kluwer Academic Publishers, Dordrecht, the Netherlands.

- E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. J. Dongarra, J. DuCroz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen (1995). *LAPACK Users' Guide*, second ed. SIAM, Philadelphia, USA.

- D. A. Andrews and L. N. Vicente (1999). Characterization of the smoothness and curvature of a marginal function for a trust-region problem. *Mathematical Programming*, **84**(1), 123–137.

**Summary.** The smoothness and curvature of a marginal function, that is, the value of the objective function as a function of the trust-region radius, for a scaled  $\ell_2$  trust-region problem are studied. The marginal function is obtained by perturbing the trust radius. The values of the marginal function and of its first and second derivatives are explicitly calculated in all possible cases. A complete study of the smoothness and curvature is given. The work is motivated by an application in statistics.

- K. Anstreicher, X. Chen, H. Wolkowicz, and Y. Yuan (1999). Strong duality for a trust-region type relaxation of the quadratic assignment problem. *Linear Algebra and Its Applications*, **301**(1–3), 121–136.

**Summary.** Quadratically constrained quadratic programs (QCPs) provide important examples of nonconvex programs. For the simple case of one quadratic constraint (the trust-region subproblem) strong duality conditions hold, as do necessary and sufficient (strengthened) second-order optimality conditions. Unfortunately, these duality results already fail for the two-trust-region subproblem. Surprisingly, there are classes of more complex, nonconvex QCPs where strong duality holds. One example is the special case of orthogonality constraints, which arise naturally in relaxations for the quadratic assignment problem (QAP). It is shown that strong duality also holds for a relaxation of QAP where the orthogonality constraint is replaced by a semidefinite inequality constraint. Using this strong duality result, and semidefinite duality, new trust-region-type necessary and sufficient optimality conditions are developed for these problems.

- K. A. Ariyawansa and D. T. M. Lau (1992). On the updating scheme in a class of collinear scaling algorithms for sparse minimization. *Journal of Optimization Theory and Applications*, **75**(1), 183–193.

**Summary.** The techniques for solving the trust-region subproblem are used for solving a variational problem arising in the updating of sparse Hessian approximations for large-scale unconstrained optimization.

- C. Ashcraft, R. G. Grimes, and J. G. Lewis (1995). Accurate symmetric indefinite linear equation solvers. Technical Report, Boeing Computer Services, Seattle, WA, USA.

- G. Auer (1993). Numerische Behandlung von Trust Region Problemen. Master's thesis, Technical University of Graz, Austria.
- O. Axelsson (1972). A generalized SSOR method. *BIT*, **12**, 443–467.
- O. Axelsson (1996). *Iterative Solution Methods*. Cambridge University Press, Cambridge, England.
- M. H. Bakr, J. W. Bandler, R. M. Biemacki, S. H. Chen, and K. Madsen (1998). A trust region aggressive space mapping algorithm for EM optimization. *IEEE Transactions on Microwave Theory and Techniques*, **46**(12, part 2), 2412–2425.
- Summary.** A new robust algorithm for electromagnetic (EM) optimization of microwave circuits is presented. The algorithm integrates a trust-region methodology with aggressive space mapping. A new automated multipoint parameter extraction process is implemented. EM optimization of a double-folded stub filter and of a high-temperature-superconductor filter are used to illustrate the approach.
- M. H. Bakr, J. W. Bandler, N. Georgieva, and K. Madsen (1999). A hybrid aggressive space mapping algorithm for EM optimization. *IEEE MTT-S International Microwave Symposium Digest*, **1**, 265–268.
- Summary.** A hybrid aggressive space-mapping optimization method, combining the trust-region aggressive space-mapping (TRASM) algorithm and direct optimization techniques, is given. No assumption that the final space-mapped design is the true optimal design is made, and the method is robust against severe misalignment between coarse and fine models. The algorithm is based on theoretical results that enable smooth switching from the TRASM optimization to direct optimization and vice versa. The algorithm is tested on several microwave filters and transformers.
- J. W. Bandler, S. H. Chen, and K. Madsen (1988). An algorithm for one-sided  $\ell_1$  optimization with application to circuit design centering. In *Proceedings 1988 IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 1795–1798, IEEE, New York, USA.
- Summary.** A highly efficient algorithm for one-sided nonlinear  $\ell_1$  optimization combines a trust-region Gauss–Newton method and a quasi-Newton method. The proposed method is used as an integral part of an approach to design centering and yield enhancement.
- T. Bannert (1994). A trust region algorithm for nonsmooth optimization. *Mathematical Programming*, **67**(2), 247–264.
- Summary.** A trust-region algorithm is proposed for minimizing the nonsmooth composite function  $F(x) = h(f(x))$ , where  $f$  is smooth and  $h$  is convex. It uses a smoothing function closely related to Fletcher's (1970a) exact differentiable penalty function. Global and local convergence to a strongly unique minimizer and to a minimizer satisfying second-order sufficiency conditions is considered.
- R. Barakat and B. H. Sandler (1992). Determination of the wave-front aberration function from measured values of the point-spread function—a 2-dimensional phase retrieval problem. *Journal of the Optical Society of America A. Optics, Image, Science, and Vision*, **9**(10), 1715–1723.
- Summary.** A method for the determination of the unknown wave-front aberration function of an optical system from noisy measurements of the corresponding point-spread function is considered. The problem is cast as an unconstrained nonlinear minimization problem. Trust-region techniques are employed for its solution using analytic expressions of the Jacobian and Hessian matrices. Illustrative numerical results are discussed.

- R. Barakat and B. H. Sandler (1999). Simultaneous determination of the modulus and phase of a coherently illuminated object from its measured diffraction image. *Journal of Optics A: Pure and Applied Optics*, **1**(5), 629–634.

**Abstract.** An algorithm for recovering the modulus and phase of a coherently illuminated object from its measured diffraction image is presented. The algorithm is based upon the fact that both the Jacobian and Hessian matrices can be evaluated exactly so that both slope and curvature information is available. The inversion problem is cast as a nonlinear unconstrained optimization problem, and trust-region techniques are employed for its solution. Representative numericals are presented.

- J. L. Barlow and G. Toraldo (1995). The effect of diagonal scaling on projected gradient methods for bound constrained quadratic programming problems. *Optimization Methods and Software*, **5**(3), 235–245.

- R. H. Bartels, G. H. Golub, and M. A. Saunders (1970). Numerical techniques in mathematical programming. In J. B. Rosen, O. L. Mangasarian, and R. Ritter, eds., *Nonlinear Programming*, pp. 123–176, Academic Press, London.

- M. C. Bartholomew-Biggs (1987). Recursive quadratic-programming methods based on the augmented Lagrangian. *Mathematical Programming Studies*, **31**, 21–41.

- M. S. Bazaraa and J. J. Goode (1982). Sufficient conditions for a globally exact penalty-function without convexity. *Mathematical Programming Studies*, **19**, 1–15.

- E. M. L. Beale (1967). Numerical methods. In J. Abadie, ed., *Nonlinear Programming*, pp. 135–205, North-Holland, Amsterdam, the Netherlands.

- B. M. Bell (1990). Global convergence of a semi-infinite optimization method. *Applied Mathematics and Optimization*, **21**, 69–88.

**Summary.** An algorithm for minimizing locally Lipschitz functions using approximate function values is presented. It yields a method for minimizing semi-infinite exact penalty functions that parallels the trust-region methods used in composite nondifferentiable optimization. A finite method for approximating a semi-infinite exact penalty function and a uniform implicit function theorem are established. An implementation and test results for the approximate penalty function are given.

- M. Bellare and P. Rogaway (1995). The complexity of approximating a nonlinear program. *Mathematical Programming*, **69**(3), 429–441.

- A. Ben-Tal and A. Nemirovskii (1997). Robust truss topology design via semidefinite programming. *SIAM Journal on Optimization*, **7**(4), 991–1016.

- A. Ben-Tal and M. Teboulle (1996). Hidden convexity in some nonconvex quadratically constrained quadratic-programming. *Mathematical Programming*, **72**(1), 51–63.

**Summary.** The minimization of an indefinite quadratic objective function subject to two-sided indefinite quadratic constraints is considered. Under a simultaneous diagonalization assumption (which trivially holds for trust-region-type problems), it is shown that the original problem is equivalent to a convex minimization problem with simple linear constraints. The special problem of minimizing a concave quadratic function subject to finitely many convex quadratic constraints is then considered and shown to be equivalent to a min-max convex

problem. In both cases, the explicit nonlinear transformations that allow the recovery of the optimal solution of the nonconvex problems via their equivalent convex counterparts are derived. Interior-point polynomial time algorithms for the solution of the equivalent convex programs are outlined.

- A. Ben-Tal and M. Zibulevsky (1997). Penalty/barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization*, **7**(2), 347–366.
- Y. Bereaux and J. R. Clermont (1997). Numerical simulation of two- and three-dimensional complex flows of viscoelastic fluids using the stream-tube method. *Mathematics and Computers in Simulation*, **44**(4), 387–400.

**Summary.** The stream-tube method in two- and three-dimensional duct flows is analysed using the concept of stream-tubes in a mapped computational domain of the physical domain, where streamlines are parallel and straight. The primary unknown of the problem includes the transformation between the two domains and the pressure. Mass conservation is automatically verified by the formulation. Memory-integral constitutive equations may be considered without the particle-tracking problem. The method is applied to flows in contractions and a three-dimensional flow involving a threefold rotational symmetry. Viscous and elastic liquids involving memory-integral equations are investigated. The discretized schemes are presented and the relevant equations solved by using optimization procedures such as the Levenberg–Morrison–Marquardt and trust-region methods.

- D. P. Bertsekas (1976). On the Goldstein-Levitin-Poljak gradient projection method. *IEEE Transactions on Automatic Control*, **AC-21**, 174–184.
- D. P. Bertsekas (1982a). *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, London.
- D. P. Bertsekas (1982b). Projected Newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, **20**(2), 221–246.
- M. J. Best (1984). Equivalence of some quadratic-programming algorithms. *Mathematical Programming*, **30**(1), 71–87.
- M. J. Best and N. Chakravarti (1990). Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, **47**, 425–439.
- M. J. Best and K. Ritter (1976). An effective algorithm for quadratic minimization problems. Technical Report 1691, University of Wisconsin, Madison, WI, USA.
- L. T. Biegler, J. Nocedal, and C. Schmid (1995). A reduced Hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, **5**(2), 314–347.
- R. H. Bielschowsky and F. A. M. Gomes (1998). Dynamical control of infeasibility in nonlinearly constrained optimization. Presentation at the Optimization ‘98 Conference, Coimbra, Portugal.

**Summary.** An algorithm for solving nonconvex problems of the form  $\min f(x)$  subject to  $h(x) = 0$  is presented. At each major iteration, a normal step  $d_v$  such that  $x_c$  satisfies  $\|h(x_c)\| = O(\|g_p(x_c)\|)$  is sought, where  $g_p(x_c)$  is the orthogonal projection of  $\nabla f(x_c)$  onto the tangent space. A tangential step  $d_h$  is then computed that reduces the Lagrangian and stays approximately tangential to the constraints. This is done by confining  $x_c + d_h$  to a cylinder

with radius  $r = O(\|g_p(x_c)\|)$  around  $h(x) = 0$ . Implementation details, as well as preliminary numerical results on problems from the CUTE collection, are presented. Global convergence results are also proved.

- M. Bierlaire (1984). HieLoW: Un logiciel d'estimation de modèles logit emboîtés. *Cahiers du MET*, **2**, 29–43.

- M. Bierlaire (1995). A robust algorithm for the simultaneous estimation of hierarchical logit models. GRT Report 95/3, Department of Mathematics, University of Namur, Belgium.

**Summary.** A trust-region method is proposed for the estimation of simultaneous hierarchical logit models, where the subproblem is solved by a truncated conjugate gradient technique. Numerical experiments indicate the power of the proposed algorithm and associated software.

- M. Bierlaire (1998). Discrete choice models. In M. Labbé, G. Laporte, K. Tanczos, and Ph. L. Toint, eds., *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, pp. 203–227, Springer-Verlag, Heidelberg, Berlin, New York.

- M. Bierlaire and Ph. L. Toint (1995). MEUSE: an origin-destination estimator that exploits structure. *Transportation Research B*, **29**(1), 47–60.

- M. Bierlaire, Ph. L. Toint, and D. Tuyttens (1991). On iterative algorithms for linear least squares problems with bound constraints. *Linear Algebra and Its Applications*, **143**, 111–143.

- M. C. Biggs (1972). Constrained minimization using recursive equality quadratic programming. In F. A. Lootsma, ed., *Numerical Methods for Nonlinear Optimization*, pp. 411–428, Academic Press, London, 1972.

- S. C. Billups and M. C. Ferris (1997). QPCOMP: A quadratic program based solver for mixed complementarity problems. *Mathematical Programming*, **76**(3), 533–562.

- Å. Björck (1996). *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, USA.

- H. G. Bock, J. P. Schlöder, and V. H. Schulz (1995). Numerik großer Differentiell-Algebraischer Gleichungen—Simulation und Optimierung. In H. Schuler, ed., *Proceßsimulation*, pp. 35–80, VCH Verlagsgesellschaft, Weinheim, Germany.

- C. Bockmann (1996). Curve-fitting and identification of physical spectra. *Journal of Computational and Applied Mathematics*, **70**(2), 207–224.

**Summary.** A modification of the trust-region Gauss–Newton method for the identification of physical spectra is described and analysed. Local convergence results are presented.

- J. M. Bofill (1995). A conjugate-gradient algorithm with a trust region for molecular-geometry optimization. *Journal of Molecular Modeling*, **1**(1), 11–17.

**Summary.** An algorithm is presented for the optimization of molecular geometries and general functions using the nonlinear conjugate gradient method with a restricted step and a restart procedure. The algorithm requires less memory storage than other conjugate gradient algorithms. Numerical results are presented, and the efficiency of the algorithm is compared with the standard conjugate gradient method. A comparison of both conjugate gradient and variable-metric methods with and without the trust-region technique is made. It is concluded that a trust region always improves the convergence. A sketch of the algorithm is given.

- P. T. Boggs, R. H. Byrd, and R. B. Schnabel (1987). A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal on Scientific and Statistical Computing*, **8**(6), 1052–1078.

**Summary.** A method for solving the orthogonal distance regression problem is described that is an analog of the trust-region Levenberg–Morrison–Marquardt algorithm. The number of unknowns involved is the number of model parameters plus the number of data points, often a very large number. By exploiting sparsity, the computational effort per step is of the same order as that required for the Levenberg–Morrison–Marquardt method for ordinary least squares. The algorithm is proved to be globally and locally convergent. Computational tests illustrate differences between orthogonal distance regression and ordinary least squares.

- P. T. Boggs, P. D. Domich, and J. E. Rogers (1995). An interior point method for general large-scale quadratic programming problems. Internal Report NISTIR 5406, Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD, USA.

- P. T. Boggs, J. R. Donaldson, R. H. Byrd, and R. B. Schnabel (1989). ORDPACK software for weighted orthogonal distance regression. *ACM Transactions on Mathematical Software*, **15**(4), 348–364.

**Summary.** ORDPACK, a software package for the weighted orthogonal distance regression problem, is described. This package is an implementation of the algorithm described in Boggs, Byrd, and Schnabel (1987) for finding the parameters that minimize the sum of the squared weighted orthogonal distances from a set of observations to a curve or surface determined by the parameters. It can also be used to solve the ordinary nonlinear least-squares problem. The package allows a general weighting scheme, provides for finite difference derivatives, and contains extensive error checking and report generating facilities.

- P. T. Boggs, A. J. Kearsley, and J. W. Tolle (1999). A practical algorithm for general large scale nonlinear optimization problems. *SIAM Journal on Optimization*, **9**(3), 755–778.

**Summary.** An effective and efficient implementation of a sequential quadratic programming algorithm for the general large-scale nonlinear programming problem is given. The quadratic programming subproblems are solved by an interior-point method and can be prematurely halted by a trust-region constraint. Numerous computational enhancements to improve the numerical performance are presented. These include a dynamic procedure for adjusting the merit function parameter and procedures for adjusting the trust-region radius. Numerical results and comparisons are presented.

- P. T. Boggs and J. W. Tolle (1989). A strategy for global convergence in a sequential quadratic programming algorithm. *SIAM Journal on Numerical Analysis*, **26**(3), 600–623.

- P. T. Boggs and J. W. Tolle (1995). Sequential quadratic programming. *Acta Numerica*, **4**, 1–51.

P. T. Boggs, J. W. Tolle, and A. J. Kearsley (1991). A merit function for inequality constrained nonlinear programming problems. Internal Report NISTIR 4702, Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD, USA.

P. T. Boggs, J. W. Tolle, and A. J. Kearsley (1994). A practical algorithm for general large scale nonlinear optimization problems. Internal Report NISTIR 5407, Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD, USA.

I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1995). CUTE: Constrained and Unconstrained Testing Environment. *ACM Transactions on Mathematical Software*, **21**(1), 123–160.

J. F. Bonnans and M. Bouhtou (1995). The trust region affine interior-point algorithm for convex and nonconvex quadratic-programming. *RAIRO Recherche Opérationnelle. Operations Research*, **29**(2), 195–217.

**Summary.** A theoretical and numerical investigation of an interior-point algorithm for quadratic programming using a trust-region scheme formulated by Ye and Tse (1989) is performed. Under a nondegeneracy hypothesis, the algorithm converges globally in the convex case. For a nonconvex problem, the sequence of points converges to a stationary point under a mild additional assumption. An asymptotic linear convergence factor that depends only on the dimension of the problem is given. Provided simple modifications are made, the method behaves well numerically.

J. F. Bonnans, J. Ch. Gilbert, C. Lemaréchal, and C. A. Sagastizábal (1995). A family of variable metric proximal methods. *Mathematical Programming, Series A*, **68**(1), 15–47.

J. F. Bonnans and G. Launay (1992). Implicit trust region algorithm for constrained optimization. Technical Report, INRIA, Le Chesnay, France.

**Summary.** Convergence of sequential quadratic programming algorithms for the nonlinear programming problems is studied in the absence of strict complementarity assumptions. Global and superlinear convergence results are obtained.

J. F. Bonnans and G. Launay (1995). Sequential quadratic-programming with penalization of the displacement. *SIAM Journal on Optimization*, **5**(4), 792–812.

J. F. Bonnans, E. Panier, A. L. Tits, and J. L. Zhou (1992). Avoiding the Maratos effect by means of a nonmonotone linesearch II. Inequality constrained problems—feasible iterates. *SIAM Journal on Numerical Analysis*, **29**, 1187–1202.

J. F. Bonnans and C. Pola (1997). A trust region interior point algorithm for linearly constrained optimization. *SIAM Journal on Optimization*, **7**(3), 717–731.

**Summary.** An extension of the trust-region quadratic programming algorithm of Ye and Tse (1989) and Bonnans and Bouhtou (1995) to nonlinear optimization subject to linear constraints is given. A linesearch is used to reduce the step if necessary. Under suitable hypotheses, the algorithm converges to a first-order stationary point. Conditions under which the unit stepsize is asymptotically accepted are analysed.

A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, **17**(1), 1–13.

J. C. Boot (1964). *Quadratic Programming*. North-Holland, Amsterdam, the Netherlands.

J. Borggaard (1994). The sensitivity equation method for optimal design. Ph.D. thesis, Department of Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

J. Borggaard and J. Burns (1997a). A PDE sensitivity equation method for optimal aerodynamic design. *Journal of Computational Physics*, **136**(2), 366–384.

**Summary.** An approach, in which the partial differential equation sensitivity equation is used to develop algorithms for computing gradients in inverse design problems but where mesh sensitivities need not be computed is considered. Moreover, when it is possible to use the computational fluid dynamics scheme for both the forward problem and the sensitivity equation, then there are further computational advantages. For a proper combination of discretization schemes, asymptotic consistency under mesh refinement is shown, which is often sufficient to guarantee convergence of the optimal design algorithm. In particular, when asymptotically consistent schemes are combined with a trust-region optimization algorithm, the resulting optimal design method converges. Such a method is described, convergence results are given, and the approach is illustrated on two optimal design problems involving shocks.

J. Borggaard and J. Burns (1997b). A sensitivity equation approach to shape optimization in fluid flows. In M. Gunzburger, ed., *Proceedings of the IMA Period of Concentration on Flow Control*, pp. 49–78, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** A sensitivity equation method is applied to shape optimization problems. An algorithm is developed and tested on a problem of designing optimal forebody simulators for a two-dimensional, inviscid supersonic flow. The algorithm uses a Broyden–Fletcher–Goldfarb–Shanno or trust-region optimization scheme, with sensitivities computed by numerically approximating the linear partial differential equations that determine the flow sensitivities. Numerical examples are presented.

J. M. Borwein (1982). Necessary and sufficient conditions for quadratic minimality. *Numerical Functional Analysis and Optimization*, **5**, 127–140.

A. Bouaricha (1997). Algorithm 765: STENMIN: a software package for large, sparse unconstrained optimization using tensor methods. *ACM Transactions on Mathematical Software*, **23**(1), 81–90.

A. Bouaricha and R. B. Schnabel (1997). Algorithm 768: TENSOLVE: A software package for solving systems of nonlinear equations and nonlinear least-squares problems using tensor methods. *ACM Transactions on Mathematical Software*, **23**(2), 174–195.

**Summary.** A modular software package for solving systems of nonlinear equations and nonlinear least-squares problems, using tensor methods, is described. It is intended for small- to medium-size problems for which it is reasonable to calculate the Jacobian or to approximate it by finite differences at each iteration. It allows the user to choose between a tensor method and a standard method based on a linear model. The tensor method approximates  $F(x)$  by

a quadratic model, where the second-order term is chosen so that the model is hardly more expensive to form, store, or solve than the linear model. The software provides both a linesearch and a two-dimensional trust-region approach. Test results indicate that tensor methods are significantly more efficient and robust than standard methods on small- and medium-sized problems in iterations and function evaluations.

- A. Bouaricha and R. B. Schnabel (1998). Tensor methods for large sparse systems of nonlinear equations. *Mathematical Programming*, **82**(3), 377–412.
- A. Bouaricha and R. B. Schnabel (1999). Tensor methods for large, sparse nonlinear least squares problems. *SIAM Journal on Scientific Computing*, **21**(4), 1199–1221.
- S. Boyd, L. El-Ghaoui, E. Feron, and V. Balakrishnan (1994). *Linear Matrix Inequalities in Systems and Control Theory*. SIAM, Philadelphia, USA.
- M. A. Branch (1995). Inexact reflective Newton methods for large-scale optimization subject to bound constraints. Technical Report TR 95-1543, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** The problem of minimizing a large-scale nonlinear function subject to simple bound constraints using the reflective Newton approach is addressed, including how it can be combined with inexact Newton techniques within a subspace trust-region method. A linesearch and a trust-region algorithm are presented that have fast convergence when negative curvature is encountered. The subspace trust-region approach is compared to other approximations to the trust-region subproblem. On problems where only positive curvature is found, these methods differ little in efficiency. For problems with negative curvature, the subspace method is more effective in capturing the negative curvature information, resulting in faster convergence. A parallel implementation on the IBM SP2 is evaluated whose scalability and efficiency are as good as the matrix-vector multiply routine it depends on.

- M. A. Branch, T. F. Coleman, and Y. Li (1999). A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, **21**(1), 1–23.

**Summary.** A subspace adaptation of the method by Coleman and Li (1996a) is proposed for solving large-scale bound-constrained minimization problems. This method can be implemented with either sparse Cholesky factorization or conjugate gradients. The convergence properties of this subspace trust-region method are as strong as those of its full-space version. Computational performance on large-scale test problems illustrates its advantages.

- L. M. Bregman (1967). The relaxation method for finding the common points of convex sets and its applications to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, **7**, 200–217.

- M. G. Breitfeld and D. F. Shanno (1994). Preliminary computational experience with modified log-barrier functions for large-scale nonlinear programming. In W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., *Large Scale Optimization: State of the Art*, pp. 45–66, Kluwer Academic Publishers, Dordrecht, the Netherlands.

- M. G. Breitfeld and D. F. Shanno (1996). Computational experience with penalty-barrier methods for nonlinear programming. *Annals of Operations Research*, **62**, 439–463.

- R. P. Brent (1973). Some efficient algorithms for solving systems of nonlinear equations. *SIAM Journal on Numerical Analysis*, **10**(2), 327–344.
- J. Brimberg and R. F. Love (1991). Estimating travel distances by the weighted  $\ell_p$  norm. *Naval Research Logistics*, **38**, 241–259.
- K. M. Brown (1969). A quadratically convergent Newton-like method based on Gaussian elimination. *SIAM Journal on Numerical Analysis*, **6**(4), 560–569.
- C. G. Broyden (1970). The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and its Applications*, **6**, 76–90.
- C. G. Broyden and N. F. Attia (1984). A smooth sequential penalty function method for nonlinear programming. In A. V. Balakrishnan and M. Thomas, eds., *Eleventh IFIP Conference on System Modelling and Optimization*, pp. 237–245, Lecture Notes in Control and Information Sciences 59, Springer-Verlag, Heidelberg, Berlin, New York.
- C. G. Broyden and N. F. Attia (1988). Penalty functions, Newton's method and quadratic programming. *Journal of Optimization Theory and Applications*, **58**(3), 377–385.
- A. G. Buckley (1978a). A combined conjugate-gradient quasi-Newton minimization algorithm. *Mathematical Programming*, **15**, 200–210.
- A. G. Buckley (1978b). Extending the relationship between the conjugate gradient and the BFGS algorithms. *Mathematical Programming*, **15**, 343–348.
- D. E. Budil, S. Lee, S. Saxena, and J. H. Freed (1996). Nonlinear-least-squares analysis of slow-motion EPR-spectra in one and two dimensions using a modified Levenberg-Marquardt algorithm. *Journal of Magnetic Resonance, Series A*, **120**(2), 155–189.
- Summary.** The application of the trust-region modification of the Levenberg–Morrison–Marquardt algorithm to the analysis of one-dimensional continuous wave (CW) electron paramagnetic resonance (EPR) and multidimensional Fourier-transform (FT) EPR spectra especially in the slow-motion regime is described. The dynamic parameters describing the motion are obtained from least-squares fitting of model calculations based on the stochastic Liouville equation (SLE) to experimental spectra. The trust-region approach is more efficient than the standard Levenberg–Morrison–Marquardt algorithm, and the efficiency of the procedure may be further increased by a separation-of-variables method in which a subset of fitting parameters is independently minimized at each iteration. An application is the fitting of multicomponent spectra, for which it is possible to obtain the relative population of each component by the separation-of-variables method. These advantages, combined with improvements in the computational solution of the SLE, have led to an order-of-magnitude reduction in computing time and have made it possible to carry out interactive, real-time fitting on a laboratory workstation. Examples are given, including multicomponent CW EPR spectra as well as two- and three-dimensional FT EPR spectra.
- A. B. Bulsari, M. Sillanpaa, and H. Saxen (1992). An expert system for continuous steel casting using neural networks. In J. L. Jamsa-Jounela and A. J. Niemi, eds., *Expert Systems in Mineral and Metal Processing. Proceedings of the IFAC Workshop*, pp. 155–159, Pergamon, Oxford, England.

**Summary.** A feedforward neural network for knowledge storage and inferencing is studied for an industrial problem. The inputs to the network receive information about an incoming ladle of steel, and the output predicts its suitability for successful continuous casting. A trust-region optimization method is used for training the network. This training method is found to be reliable and robust.

- J. P. Bulteau and J. P. Vial (1983). Unconstrained optimization by approximation of a projected gradient path. CORE Discussion Paper 8352, CORE, UCL, Louvain-la-Neuve, Belgium.

**Summary.** Bulteau and Vial (1987) discuss a general algorithm based on a one-dimensional search over a curvilinear path according to a trust-region scheme. An implementation using an approximation of the projected gradient path on a two-dimensional space is given. This algorithm is endowed with attractive convergence properties. Newton- and quasi-Newton-like variants are discussed, with corresponding numerical experiments.

- J. P. Bulteau and J. P. Vial (1985). A restricted trust region algorithm for unconstrained optimization. *Journal of Optimization Theory and Applications*, **47**(4), 413–435.

**Summary.** An efficient implementation of a trust-region method is proposed, in which the trust region is restricted to an appropriately chosen two-dimensional subspace. Convergence properties are discussed and numerical results are reported.

- J. P. Bulteau and J. P. Vial (1987). Curvilinear path and trust region in unconstrained optimization—a convergence analysis. *Mathematical Programming Studies*, **30**, 82–101.

**Summary.** A general algorithm for unconstrained optimization is proposed. Its basic step consists in finding a “good” successor point to the current iterate by choosing it along a curvilinear path within a trust region. Properties that an arbitrary path should satisfy in order to achieve global convergence and fast asymptotic convergence are given. Various paths that have been proposed in the literature are reviewed in this light.

- D. S. Bunch, D. M. Gay, and R. E. Welsch (1993). Algorithm 717: Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. *ACM Transactions on Mathematical Software*, **19**(1), 109–130.

**Summary.** Fortran 77 subroutines are presented that solve statistical parameter estimation problems for general nonlinear models, e.g., nonlinear least squares, maximum likelihood, maximum quasi-likelihood, generalized nonlinear least squares, and some robust fitting problems. The basic method, a generalization of the NL2SOL algorithm for nonlinear least squares, employs a trust-region scheme for computing trial steps, maintains a secant approximation to the second-order part of the Hessian, and adaptively switches between Gauss–Newton and full Newton approximations. Gauss–Newton steps are computed using a corrected seminormal equations approach. The subroutines include variants that handle simple bounds on the parameters and that compute approximate regression diagnostics.

- J. R. Bunch (1974). Partial pivoting strategies for symmetric matrices. *SIAM Journal on Numerical Analysis*, **11**, 521–528.

- J. R. Bunch and L. C. Kaufman (1977). Some stable methods for calculating inertia and solving symmetric linear equations. *Mathematics of Computation*, **31**, 163–179.

- J. R. Bunch and L. C. Kaufman (1980). A computational method for the indefinite quadratic programming problem. *Linear Algebra and Its Applications*, **34**, 341–370.
- J. R. Bunch and B. N. Parlett (1971). Direct methods for solving symmetric indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, **8**(4), 639–655.
- J. V. Burke (1990). On the identification of active constraints II: The nonconvex case. *SIAM Journal on Numerical Analysis*, **27**(4), 1081–1102.
- J. V. Burke (1992). A robust trust region method for constrained nonlinear programming problems. *SIAM Journal on Optimization*, **2**(2), 324–347.

**Summary.** A general framework for trust-region algorithms for constrained problems is presented that does not require strong regularity hypotheses and allows very general constraints. The approach is modeled on the one given by Powell for convex composite optimization problems and is driven by linear subproblems that yield viable estimates for the value of an exact penalty parameter. These results are applied to the Wilson–Han–Powell sequential quadratic programming algorithm and Fletcher’s sequential  $\ell_1$  quadratic programming algorithm. Local convergence results are given.

- J. V. Burke and S. P. Han (1989). A robust sequential quadratic-programming method. *Mathematical Programming*, **43**(3), 277–303.
- J. V. Burke and J. J. Moré (1988). On the identification of active constraints. *SIAM Journal on Numerical Analysis*, **25**(5), 1197–1211.
- J. V. Burke and J. J. Moré (1994). Exposing constraints. *SIAM Journal on Optimization*, **4**(3), 573–595.
- J. V. Burke, J. J. Moré, and G. Toraldo (1990). Convergence properties of trust region methods for linear and convex constraints. *Mathematical Programming, Series A*, **47**(3), 305–336.
- Summary.** A convergence theory is developed for convex and linearly constrained trust-region methods which only requires that the step between iterates produce a sufficient reduction in the subproblem. Global convergence is established for a general convex problem while local analysis is for linearly constrained problems. It is shown that if the sequence converges to a nondegenerate stationary point then the active constraints at the solution are identified in a finite number of iterations. As a consequence, rate-of-convergence results are developed by assuming that the step is a truncated Newton method. This development is mainly geometrical; such an approach allows the development of a convergence theory without any linear independence assumptions.
- J. V. Burke and A. Weigmann (1997). Notes on limited memory BFGS updating in a trust-region framework. Department of Mathematics, University of Washington, Seattle, WA, USA.
- Summary.** A limited-memory Broyden–Fletcher–Goldfarb–Shanno method is described that uses rescaling at each iteration and a trust-region technique to ensure convergence. The effects of a nonmonotone technique as well as that of an implicit scheme for updating the trust-region radius are discussed. Numerical experiments are reported.

- J. C. Butcher, Z. Jackiewicz, and H. D. Mittelmann (1997). A nonlinear optimization approach to the construction of general linear methods of high order. *Journal of Computational and Applied Mathematics*, **81**(2), 181–196.

**Summary.** The construction of diagonally implicit multistage integration methods of order and stage order  $p = q = 7$  and  $p = q = 8$  for ordinary differential equations is described. These methods were obtained using variable-model trust-region least-squares algorithms.

- R. H. Byrd (1990). On the convergence of constrained optimization methods with accurate Hessian information on a subspace. *SIAM Journal on Numerical Analysis*, **27**(1), 141–153.

- R. H. Byrd (1999). Step computation in a trust region interior point method. Presentation at the First Workshop on Nonlinear Optimization “Interior-Point and Filter Methods”, Coimbra, Portugal.

**Summary.** Approximate methods for computing a trust-region step in an interior-point method for nonlinear inequality constrained optimization are considered, and some new approaches proposed. The extent to which they provide the benefits promised by trust regions in the cases of negative curvature and rank deficiency are discussed.

- R. H. Byrd, J. Ch. Gilbert, and J. Nocedal (1996). A trust region method based on interior point techniques for nonlinear programming. Technical Report 2896, INRIA, Rocquencourt, France.

**Summary.** An algorithm for minimizing a nonlinear function subject to nonlinear equality and inequality constraints is described. It can be seen as an extension of primal interior-point methods to nonconvex optimization. The algorithm applies sequential quadratic programming techniques to a sequence of barrier problems and uses trust regions to ensure the robustness of the iteration and to allow the direct use of second-order derivatives. A convergence analysis is presented.

- R. H. Byrd, M. E. Hribar, and J. Nocedal (2000). An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*, **9**(4), 877–900.

**Summary.** An algorithm for solving large nonlinear programming problems is described. It incorporates sequential quadratic programming and trust-region techniques within the interior-point method. Sequential quadratic programming ideas are used to efficiently handle nonlinearities in the constraints. Trust-region strategies allow the algorithm to treat convex and nonconvex problems uniformly, permit the direct use of second derivative information, and provide a safeguard in the presence of nearly dependent constraint gradients. Both primal and primal-dual versions of the algorithm are developed, and their performance is compared with that of LANCELOT on a set of large and difficult nonlinear problems.

- R. H. Byrd, H. F. Khalfan, and R. B. Schnabel (1996). Analysis of a symmetric rank-one trust region method. *SIAM Journal on Optimization*, **6**(4), 1025–1039.

**Summary.** Computational studies have considered the symmetric rank-1 (SR1) method for unconstrained optimization and shown that the method has an  $(n+1)$ -step  $q$ -superlinear rate of convergence. The proposed analysis makes neither of the assumptions of uniform linear independence of the iterates nor positive definiteness of the Hessian approximations that have been made in former such analyses. The trust-region method is standard but requires the Hessian approximation to be updated after all steps, including rejected ones. Computational results indicate that this feature, safeguarded in a way that is consistent with the convergence analysis, does not harm the efficiency of the SR1 trust-region method.

- R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, **16**(5), 1190–1208.
- R. H. Byrd and J. Nocedal (1991). An analysis of reduced Hessian methods for constrained optimization. *Mathematical Programming*, **49**(3), 285–323.
- R. H. Byrd and R. B. Schnabel (1986). Continuity of the null space basis and constrained optimization. *Mathematical Programming*, **35**(1), 32–41.
- R. H. Byrd, R. B. Schnabel, and G. A. Shultz (1987). A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, **24**, 1152–1170.
- Summary.** A trust-region-based method is given for a general nonlinearly constrained optimization problem. It iteratively minimizes a quadratic model of the Lagrangian subject to a possibly relaxed linearization of the problem constraints and a trust-region constraint. The model minimization may be done approximately with a dogleg-type approach. Global convergence is shown. A second-order correction step is also described. If sufficiently precise Hessian information is used, this step ensures locally quadratic convergence and satisfaction of the second-order necessary conditions. An example shows that, without this correction, a situation similar to the Maratos (1978) effect may occur where the iteration is unable to move away from a saddle point.
- R. H. Byrd, R. B. Schnabel, and G. A. Shultz (1988). Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming*, **40**(3), 247–263.
- Summary.** Computational results are given, showing that the two-dimensional minimization approach of Byrd, Schnabel, and Shultz (1987) gives nearly optimal reductions in the  $n$ -dimensional quadratic model over a wide range of test cases. It is shown that there is very little difference, in efficiency and reliability, between using the approximate or exact trust-region step when solving standard test problems for unconstrained optimization.
- R. H. Byrd, R. A. Tapia, and Y. Zhang (1992). An SQP augmented Lagrangian BFGS algorithm for constrained optimization. *SIAM Journal on Optimization*, **2**(2), 210–241.
- P. H. Calamai and J. J. Moré (1987). Projected gradient methods for linearly constrained problems. *Mathematical Programming*, **39**, 93–116.
- T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno (1993). Higher-order predictor-corrector interior point methods with application to quadratic objectives. *SIAM Journal on Optimization*, **3**(4), 696–725.
- C. W. Carroll (1959). An operations research approach to the economic optimization of a kraft pulping process. Ph.D. thesis, Institute of Paper Chemistry, Appleton, WI, USA.
- C. W. Carroll (1961). The created response surface technique for optimizing nonlinear restrained systems. *Operations Research*, **9**(2), 169–184.

- R. G. Carter (1986). Multi-model algorithms for optimization. Technical Report TR86-3, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A trust-region formulation for multimodel methods is presented which allows the efficient incorporation of an arbitrary number of models. Global convergence is established for three classes of algorithms under different assumptions on the models. Firstly, essentially any multimodel algorithm is globally convergent if each of the models is sufficiently well behaved. Secondly, algorithms based on the central feature of the NL2SOL switching system are globally convergent if one model is well behaved and each other model obeys a “sufficient predicted decrease” condition. No requirement is made that these alternate models be quadratic. Finally, algorithms of the second type that directly enforce the “sufficient predicted decrease” condition are globally convergent if a single model is sufficiently well behaved.

- R. G. Carter (1987). Safeguarding Hessian approximations in trust region algorithms. Technical Report TR87-12, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** It is shown that the assumptions on the Hessian approximations in a trust-region method for unconstrained optimization can be replaced by a uniform upper bound on the sequence of Rayleigh quotients of the Hessian approximations in the gradient directions. This suggests both a simple procedure for detecting questionable approximations and several natural procedures for correcting them when detected. In numerical tests, one of these procedures increased the reliability of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method by a factor of 2. For those problems where both the safeguarded and original methods were successful, this safeguarding procedure improved the average efficiency of the BFGS by 10 to 20 percent.

- R. G. Carter (1991). On the global convergence of trust region methods using inexact gradient information. *SIAM Journal on Numerical Analysis*, **28**(1), 251–265.

**Summary.** Strong global convergence results are demonstrated for trust-region methods for unconstrained minimization where the gradient values are approximated rather than computed exactly, provided they obey a simple relative error condition. No requirement is made that gradients be recomputed to successively greater accuracy after unsuccessful iterations.

- R. G. Carter (1993). Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information. *SIAM Journal on Scientific and Statistical Computing*, **14**(2), 368–388.

**Summary.** The computational performance of trust-region methods for nonlinear optimization is investigated for cases when high-accuracy evaluations of function and gradient are unavailable or prohibitively expensive, and theoretical predictions that such methods are convergent even with relative gradient errors of 0.5 or more is confirmed. The proper choice of the amount of accuracy to use in function and gradient evaluations can result in orders-of-magnitude savings in computational cost.

- L. Case (1997). An  $\ell_1$  penalty function approach to the nonlinear bilevel programming problem. Ph.D. thesis, University of Waterloo, ON, Canada.

**Summary.** The nonlinear bilevel problem is a difficult constrained optimization problem in which the variables are partitioned into two sets,  $z$  and  $y$ . The feasibility conditions require that  $y$  be the solution of a separate optimization problem. The original problem is replaced by stating the necessary conditions for a solution and determining a one-level programming problem using an exact penalty function to attempt to satisfy these conditions. The resulting nonconvex, nonsmooth problems are solved by a trust-region approach and specialized techniques are used to overcome difficulties arising from the nondifferentiability. A unique method

- is developed to handle degeneracy. Proof of convergence to a minimum of the penalty function is given. Test results and an analysis of the solutions are included.
- A. Cauchy (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences*, pp. 536–538.
- M. R. Celis (1985). A trust region strategy for nonlinear equality constrained optimization. Technical Report TR85-4, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.
- Summary.** An approach for equality constrained optimization problems based on a trust-region strategy is presented. The direction selected is not necessarily the solution of the standard quadratic programming subproblem.
- M. R. Celis, J. E. Dennis, and R. A. Tapia (1985). A trust region strategy for nonlinear equality constrained optimization. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., *Numerical Optimization* 1984, pp. 71–82, SIAM, Philadelphia, USA.
- Summary.** Same as for Celis (1985).
- A. Cesar, H. Agren, T. Helgaker, P. Jorgensen, and H. J. A. Jensen (1991). Excited-state structures and vibronic spectra of  $\text{H}_2\text{CO}_+$ ,  $\text{HDCO}_+$ , and  $\text{D}_2\text{CO}_+$  using molecular gradient and Hessian techniques. *Journal of Chemical Physics*, **95**(8), 5906–5917.
- Summary.**  $\text{H}_2\text{CO}_+$  and its deuterated species are chosen to demonstrate the potential for using second-order multiconfigurational self-consistent field theory to optimize structures and calculate properties of ionized and excited states. The focus is on calculation of multidimensional vibronic spectra using only the molecular energy, gradient, and Hessian of the potential hypersurface. Second-order multiconfigurational self-consistent field optimization on lowest excited states using the trust-region algorithm is found to give the same stable convergence as for neutral ground states. For higher lying states, the problem of multidimensional potential crossings renders the calculations more difficult.
- Y. Chabrillac and J.-P. Crouzeix (1984). Definiteness and semidefiniteness of quadratic forms revisited. *Linear Algebra and Its Applications*, **63**, 283–292.
- R. M. Chamberlain, M. J. D. Powell, C. Lemaréchal, and H. C. Pedersen (1982). The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Studies*, **16**, 1–17.
- T. F. Chan, J. A. Olkin, and D. W. Cooley (1992). Solving quadratically constrained least squares using black box solvers. *BIT*, **32**, 481–495.
- R. Chandra (1978). Conjugate gradient methods for partial differential equations. Ph.D. thesis, Yale University, New Haven, CT, USA.
- Y. Y. Chang and R. W. Cottle (1980). Least-index resolution of degeneracy in quadratic programming. *Mathematical Programming*, **18**(2), 127–137.
- C. Charalambous (1978). A lower bound for the controlling parameter of the exact penalty functions. *Mathematical Programming*, **15**(3), 278–290.

- C. Charalambous (1979). Acceleration of the least  $p$ -th algorithm for minimax optimization with engineering applications. *Mathematical Programming*, **17**(1), 270–297.
- B. Chen, X. Chen, and Ch. Kanzow (1997). A penalized Fischer-Burmeister NCP-function: Theoretical investigation and numerical results. Technical Report A-126, Institute of Applied Mathematics, University of Hamburg, Germany.
- C. Chen and O. L. Mangasarian (1996). A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications*, **5**(2), 97–138.
- L. Chen, N. Deng, and J. Zhang (1995). A trust region method with partial-update technique for unary optimization. In D. Z. Du, X. S. Zhang, and K. Cheng, eds., *Operations Research and Its Applications. Proceedings of the First International Symposium, ISORA '95*, pp. 40–46, Beijing World Publishing, Beijing, China.
- Summary.** A modified partial-update algorithm for solving unconstrained unary optimization problems is proposed, based on trust-region stabilization via indefinite dogleg curves. This algorithm only partially updates an approximation to the Hessian matrix in each iteration by applying a limited number of rank-1 updates to its Bunch–Parlett factorization. In contrast with the original algorithms proposed by Goldfarb and Wang (1993), the algorithm not only converges globally but also possesses a locally quadratic convergence rate. Furthermore, numerical experiments show improved performance.
- L. Chen, N. Deng, and J. Zhang (1998). Modified partial-update Newton-type algorithms for unary optimization. *Journal of Optimization Theory and Applications*, **97**(2), 385–406.
- Summary.** Two modified partial-update algorithms for solving unconstrained unary optimization problems based on trust-region stabilization via indefinite dogleg curves are proposed. They both partially update an approximation to the Hessian matrix in each iteration by using the symmetric rank-1 updating of the Bunch–Parlett factorization. They converge globally with a locally quadratic or superlinear convergence rate. Numerical experiments indicate that they outperform the trust-region method, which uses some other partial update criteria.
- X. D. Chen and Y. Yuan (1999). On local solutions of the Celis–Dennis–Tapia subproblem. *SIAM Journal on Optimization*, **10**(2), 359–383.
- Summary.** The distribution of the local solutions of the Celis–Dennis–Tapia (CDT) subproblem that appears in some trust-region algorithms for nonlinear optimization is discussed. Examples illustrate the differences between the CDT subproblem and the single-ball constraint subproblem. The complexity of the CDT subproblem is shown not to depend on the complexity of the structure of the dual plane, which opens the possibility of searching the global minimizer in this plane.
- Z. Chen (1995). A new trust region algorithm for optimization with simple bounds. In D. Z. Du, X. S. Zhang and K. Cheng, eds., *Operations Research and Its Applications. Proceedings of the First International Symposium, ISORA '95*, pp. 49–58, Beijing World Publishing, Beijing, China.
- Summary.** A globally convergent trust-region algorithm is presented for minimizing a differentiable function of many variables with simple bounds. It is proved that the correct active set is identified in a finite number of iterations under a strict complementarity condition, allowing a fast asymptotic rate of convergence.

Z. Chen (1996). Some algorithms for a class of CDT subproblems. In D. Du, X. Zhang, and W. Wang, eds., *Lecture Notes in Operations Research*, pp. 108–114, Beijing World Publishing, Beijing, China.

Z. W. Chen and J. Y. Han (1996). A trust region algorithm for optimization with nonlinear equality and linear inequality constraints. *Science in China Series A — Mathematics Physics Astronomy*, **39**(8), 799–806.

**Summary.** A globally convergent trust-region algorithm is presented to minimize a smooth function of many variables with nonlinear equality and linear inequality constraints.

S. H. Cheng and N. J. Higham (1998). A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM Journal on Matrix Analysis and Applications*, **19**(4), 1097–1110.

Y. B. Cheng and J. K. Sykulski (1996). Automated design and optimization of electromechanical actuators. *International Journal of Numerical Modelling—Electronic Networks Devices and Fields*, **9**(1–2), 59–69.

**Summary.** The suitability of various optimization techniques for the magnetic design of electromechanical actuators is examined. The Gauss–Newton, Levenberg–Morrison–Marquardt (LMM), and trust-region algorithms are compared using 18 test functions. The LMM method is chosen for its robustness and fast convergence and incorporated into an automated computer-aided design optimization system (EAMON). As examples, a direct current solenoid actuator with truncated cone pole face is optimized to produce a user-specified force-displacement characteristic, and an actuator solenoid is optimized to produce maximum energy per stroke.

C. M. Chin and R. Fletcher (1999). Convergence properties of SLP-filter algorithms that use EQP steps. Numerical Analysis Report, Department of Mathematics, University of Dundee, Scotland.

T. T. Chow and P. K. Chen (1994). A new trust region global strategy for unconstrained optimization. In 1994 *International Computer Symposium Conference Proceedings. National Chiao Tung University, Hsinchu, Taiwan*, Vol. 1, pp. 394–401.

**Summary.** This paper introduces the tensor dogleg method, a trust-region technique for solving unconstrained optimization problems, which seems to outperform the linesearch-based TENMIN package.

V. Chvátal (1983). *Linear Programming*. W. H. Freeman, New York, San Francisco.

F. H. Clarke (1983). *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts, Wiley, Chichester, England. Reprinted as Classics in Applied Mathematics 5, SIAM, Philadelphia, USA, 1990.

J. R. Clermont, M. E. Delalande, T. Pham Dinh, and A. Yassine (1991). Analysis of plane and axisymmetrical flows of incompressible fluids with the stream tube method—numerical-simulation by trust-region optimization algorithm. *International Journal for Numerical Methods in Fluids*, **13**(3), 371–399.

**Summary.** Concepts for the study of incompressible plane or axisymmetric flows are analysed by the stream-tube method. Flows without eddies and pure vortex flows are considered in a

transformed domain where the mapped streamlines are rectilinear or circular. The transformation between the physical domain and the computational domain is an unknown of the problem. A trust-region algorithm is given for solving the relevant nonlinear set of equations; experimental results show that it is more robust than the Newton–Raphson method.

- A. K. Cline, A. R. Conn, and C. F. Van Loan (1982). Generalizing the LINPACK condition estimator. In J. P. Hennart, ed., *Numerical Analysis*, pp. 73–83, Lecture Notes in Mathematics 909, Springer-Verlag, Heidelberg, Berlin, New York.
- A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson (1979). An estimate for the condition number of a matrix. *SIAM Journal on Numerical Analysis*, **16**, 368–375.
- T. F. Coleman (1994). Linearly constrained optimization and projected preconditioned conjugate gradients. In J. Lewis, ed., *Proceedings of the Fifth SIAM Conference on Applied Linear Algebra*, pp. 118–122, SIAM, Philadelphia, USA.
- T. F. Coleman, M. A. Branch, and A. Grace (1999). *Optimization Toolbox* 2.0. The MathWorks Inc., Natick, MA, USA.
- T. F. Coleman and A. R. Conn (1980). Second-order conditions for an exact penalty function. *Mathematical Programming*, **19**(2), 178–185.
- T. F. Coleman and A. R. Conn (1982a). Non-linear programming via an exact penalty-function: Asymptotic analysis. *Mathematical Programming*, **24**(2), 123–136.
- T. F. Coleman and A. R. Conn (1982b). Non-linear programming via an exact penalty-function: Global analysis. *Mathematical Programming*, **24**(2), 137–161.
- T. F. Coleman and P. A. Fenyes (1992). Partitioned quasi-Newton methods for non-linear equality constrained optimization. *Mathematical Programming*, **53**(1), 17–44.
- T. F. Coleman and C. Hempel (1990). Computing a trust region step for a penalty function. *SIAM Journal on Scientific and Statistical Computing*, **11**(1), 180–201.
- Summary.** The minimization of a quadratic function subject to an ellipsoidal constraint is considered in the case when the matrix involved is the Hessian of a penalty function  $p(x) = f(x) + (1/2\mu)c(x)^T c(x)$ . Most applications require  $p(x)$  to be minimized for values of  $\mu$  decreasing to zero. The algorithm of Moré and Sorensen (1983) is modified so as to be less sensitive to the nature of finite-precision arithmetic in this situation. Numerical experiments illustrate the stability of the modified algorithm.
- T. F. Coleman and L. A. Hulbert (1989). A direct active set algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming, Series B*, **45**(3), 373–406.
- T. F. Coleman and L. A. Hulbert (1993). A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds. *SIAM Journal on Optimization*, **3**(2), 298–321.

T. F. Coleman and Y. Li (1994). On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, **67**(2), 189–224.

T. F. Coleman and Y. Li (1996a). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, **6**(2), 418–445.

**Summary.** A trust-region approach for minimizing a nonlinear function subject to simple bounds is proposed that does not require that a quadratic programming subproblem with inequality constraints be solved every iteration. Instead, a solution to a trust-region subproblem is sought. The iterates generated are strictly feasible. The proposed method reduces to a standard trust-region approach for the unconstrained problem. Global and locally quadratic convergence is established. Preliminary numerical experiments are reported.

T. F. Coleman and Y. Li (1996b). A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, **6**(4), 1040–1058.

T. F. Coleman and Y. Li (1997). A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints. Technical Report TR 97-1642, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** Global and local convergence properties of the trust-region and affine-scaling interior-point method (TRAM) by Coleman and Li (1998a) are established. It is shown that a trust-region solution is asymptotically in the interior of the trust-region subproblem and that a damped trust-region step can achieve quadratic convergence.

T. F. Coleman and Y. Li (1998a). Combining trust region and affine scaling for linearly constrained nonconvex minimization. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 219–250, Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** An interior-point method is proposed for nonconvex minimization with linear inequality constraints. It combines the trust-region idea for nonlinearity and affine-scaling technique for constraints, and ensures that the original objective function is monotonically decreased. A subproblem is formed that asymptotically yields an approximate Newton step, directly derived from the complementarity conditions. Global convergence is achieved by possibly using trust regions with different shapes. A reflection technique accelerates convergence. Explicit sufficient decrease conditions are proposed. Computational results of a two-dimensional implementation are reported for large-scale problems.

T. F. Coleman and Y. Li (1998b). A primal-dual trust region algorithm for nonconvex programming using a  $\ell_1$  penalty function. Presentation at the Optimization ‘98 Conference, Coimbra, Portugal.

**Summary.** A primal-dual algorithm is proposed for nonconvex programming. Primal and dual steps are derived directly from the complementarity conditions. A primal trust-region step is used to yield decrease for the  $\ell_1$  penalty function, and a dual constrained least-squares step yields decrease for an appropriate function of dual variables. Reflection procedures are used to accelerate convergence, and preliminary computational results are reported.

T. F. Coleman and Y. Li (2000). An affine scaling trust region algorithm for nonlinear programming. Technical report, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** Typical interior-point algorithms for nonconvex, or even convex, programming problems do not yield monotonic improvement of the objective function value. A monotonic affine scaling trust-region algorithm, using an exact  $\ell_1$  penalty function, is proposed for non-convex programming. Affine scaling Newton steps are derived directly from the complementarity conditions. A primal trust-region subproblem is proposed for globalization. A dual subproblem is formulated to facilitate dual variables updates; its solution yields decrease of the  $\ell_1$  function. Global convergence of the proposed algorithm is established.

- T. F. Coleman and A. Liao (1995). An efficient trust region method for unconstrained discrete-time optimal control problem. *Computational Optimization and Applications*, **4**(1), 47–66.

**Summary.** A method is proposed that incorporates the trust-region idea with the local stage-wise Newton's method for discrete-time optimal control (DTOC) problems. This method has strong global and local convergence properties yet remains economical. Preliminary numerical results illustrate the behaviour of the algorithm. Some DTOC problems that have appeared in the literature are collected in an appendix.

- T. F. Coleman and J. Liu (1999). An interior Newton method for quadratic programming. *Mathematical Programming, Series A*, **85**(3), 491–524.

**Summary.** An interior-point method is proposed for the general quadratic programming problem. The method converges globally to a point satisfying the second-order necessary optimality conditions, and the rate of convergence is two-step quadratic if the limit point is a strong minimizer. Preliminary numerical experiments indicate that the method has practical potential.

- T. F. Coleman and J. J. Moré (1983). Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal on Numerical Analysis*, **20**, 187–209.

- T. F. Coleman and J. J. Moré (1984). Estimation of sparse Hessian matrices and graph coloring problems. *Mathematical Programming*, **28**, 243–270.

- T. F. Coleman and P. E. Plassman (1992). A parallel nonlinear least-squares solver— theoretical analysis and numerical results. *SIAM Journal on Scientific and Statistical Computing*, **13**(3), 771–793.

- T. F. Coleman and A. Pothen (1986). The null space problem I. Complexity. *SIAM Journal on Algebraic and Discrete Methods*, **7**(4), 527–537.

- T. F. Coleman and A. Pothen (1987). The null space problem II. Algorithms. *SIAM Journal on Algebraic and Discrete Methods*, **8**(4), 544–563.

- T. F. Coleman and D. C. Sorensen (1984). A note on the computation of an orthonormal basis for the null space of a matrix. *Mathematical Programming*, **29**(2), 234–242.

- T. F. Coleman and W. Yuan (1995). A new trust region algorithm for equality constrained optimization. Technical Report TR95-1477, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** A trust-region algorithm for solving nonlinear equality constrained problems is presented. At each iterate a change of variables improves the ability of the algorithm to follow the constraint level sets. The algorithm employs quadratic penalty functions to obtain global convergence. It converges globally and Q-quadratically to a point satisfying second-order necessary optimality conditions. Preliminary numerical experiments are presented.

- L. Collatz (1966). *Functional Analysis and Numerical Mathematics*. Academic Press, London.
- B. Colson (1999). Mathematical programs with equilibrium constraints and nonlinear bilevel programming problems. Master's thesis, Department of Mathematics, University of Namur, Belgium.
- P. Concus, G. H. Golub, and D. P. O'Leary (1976). Numerical solution of nonlinear elliptic partial differential equations by a generalized conjugate gradient method. In J. Bunch and D. Rose, eds., *Sparse Matrix Computations*, pp. 309–332, Academic Press, London.
- A. R. Conn (1973). Constrained optimization via a nondifferentiable penalty function. *SIAM Journal on Numerical Analysis*, **10**(4), 760–779.
- A. R. Conn and C. Charalambous (1975). Optimization of microwave networks. *IEEE Transactions on Microwave Theory and Techniques*, **23**(10), 834–838.
- A. R. Conn, P. K. Coulman, R. A. Haring, G. L. Morrill, and C. Visweswarah (1996). Optimization of custom MOS circuits by transistor sizing. In *IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers (Cat. No. 96CB35991)*, pp. 174–180, IEEE Computer Society Press, Los Alamitos, CA, USA.

**Summary.** JiffyTune is a circuit optimization tool that automates the tuning task. Delay, rise/fall time, area, and power targets are accommodated. Each (weighted) target can be either a constraint or an objective function. Min-max optimization is supported. Transistors can be ratioed and similar structures grouped to ensure regular layouts. Bounds on transistor widths are supported. JiffyTune uses the trust-region-based package **LANCELOT**. In the inner loop of the optimization, the fast circuit simulator SPECS is used to evaluate the circuit. SPECS is unique in its ability to provide time-domain sensitivities, thereby enabling gradient-based optimization. Both the adjoint and direct methods of sensitivity computation have been implemented in SPECS. Interfaces in the Cadence and SLED design systems have been constructed.

- A. R. Conn, N. I. M. Gould, M. Lescrenier, and Ph. L. Toint (1994). Performance of a multifrontal scheme for partially separable optimization. In S. Gomez and J. P. Hennart, eds., *Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, Vol. 275, pp. 79–96, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint (2000). A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, **87**(2), 215–249.

**Summary.** A primal-dual algorithm is proposed for the minimization of nonconvex objective functions subject to simple bounds and linear equality constraints. The method uses a primal-dual trust-region model to ensure descent on a suitable merit function. Convergence of a well-defined subsequence of iterates is proved to a second-order critical point from arbitrary starting points. Algorithmic variants are discussed and preliminary numerical results presented.

- A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint (1993). Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints. *SIAM Journal on Optimization*, **3**(1), 164–221.

**Summary.** Trust-region algorithms for the solution of nonlinear optimization problems with a convex feasible set are presented. The theory given allows for the use of general norms. Furthermore, the proposed algorithms do not require the explicit computation of the projected gradient and can therefore be adapted to cases where the projection onto the feasible domain may be expensive to calculate. Strong global convergence results are derived. The linear and nonlinear constraints that are binding at the solution are identified by the algorithms in a finite number of iterations.

- A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint (1996a). Convergence properties of an augmented Lagrangian algorithm for optimization with a combination of general equality and linear constraints. *SIAM Journal on Optimization*, **6**(3), 674–703.

- A. R. Conn, N. I. M. Gould, A. Sartenaer, and Ph. L. Toint (1996b). Convergence properties of minimization algorithms for convex constraints using a structured trust region. *SIAM Journal on Optimization*, **6**(4), 1059–1086.

**Summary.** A class of structured trust-region algorithms is presented for minimization problems within convex feasible regions, in which the structure of the problem is explicitly used in the definition of the trust region. Global convergence is established.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1988a). Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM Journal on Numerical Analysis*, **25**(2), 433–460. (See also same journal, **26**(3), 764–767, 1989.)

**Summary.** The global convergence properties of trust-region algorithms for unconstrained optimization are extended to the case where bounds on the variables are present. Weak conditions on the accuracy of the Hessian approximations are considered. When the strict complementarity condition holds, the proposed algorithms reduce to an unconstrained calculation after finitely many iterations, allowing fast convergence.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1988b). Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation*, **50**, 399–430.

**Summary.** The results of tests on the trust-region methods proposed by Conn, Gould, and Toint (1988a) for solving the bound-constrained minimization problem are discussed.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1990). An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project. In R. Glowinski and A. Lichnewsky, eds., *Computing Methods in Applied Sciences and Engineering*, pp. 42–51, SIAM, Philadelphia, USA.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1991a). Convergence of quasi-Newton matrices generated by the symmetric rank one update. *Mathematical Programming*, **50**(2), 177–196.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1991b). A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, **28**(2), 545–572.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1992a). Intensive numerical tests with **LANCELOT** (Release A): The complete results. Technical Report 92/15, Department of Mathematics, University of Namur, Belgium.

**Summary.** The detailed results for the tests reported by Conn, Gould, and Toint (1996) are presented.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1992b). *LANCELOT: A Fortran package for Large-Scale Nonlinear Optimization (Release A)*. Springer Series in Computational Mathematics, Springer-Verlag, Heidelberg, Berlin, New York.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1992c). On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear equality constraints and simple bounds. In D. F. Griffiths and G. A. Watson, eds., *Numerical Analysis 1991*, pp. 49–68, Pitman Research Notes in Mathematics Series 260, Longman Scientific & Technical, Harlow, Essex, England.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1994). A note on using alternative second-order models for the subproblems arising in barrier function methods for minimization. *Numerische Mathematik*, **68**, 17–33.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1996). Numerical experiments with the **LANCELOT** package (Release A) for large-scale nonlinear optimization. *Mathematical Programming, Series A*, **73**(1), 73–110.

**Summary.** The algorithmic options available within Release A of **LANCELOT**, a trust-region-based Fortran package for large-scale nonlinear optimization, are presented. The results of intensive numerical tests are described, and the relative merits of the options discussed. The experiments described involve both academic and applied problems. Conclusions specific to **LANCELOT** and of more general scope are made.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1997a). A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Mathematics of Computation*, **66**, 261–288.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1997b). Methods for nonlinear constraints in optimization calculations. In I. Duff and A. Watson, eds., *The State of the Art in Numerical Analysis*, pp. 363–390, Oxford University Press, Oxford, England.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1997c). On the number of inner iterations per outer iteration of a globally convergent algorithm for optimization with general nonlinear inequality constraints and simple bounds. *Computational Optimization and Applications*, **7**(1), 41–70.

- A. R. Conn, N. I. M. Gould, and Ph. L. Toint (1999). A primal-dual algorithm for minimizing a nonconvex function subject to bound and linear equality constraints. In G. Di Pillo and F. Giannessi, eds., *Nonlinear Optimization and Applications 2*, pp. 15–30, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- A. R. Conn and Y. Li (1992). A structure-exploiting algorithm for nonlinear minimax problems. *SIAM Journal on Optimization*, **2**(2), 242–263.
- A. R. Conn and T. Pietrzykowski (1977). A penalty function method converging directly to a constrained optimum. *SIAM Journal on Numerical Analysis*, **14**(2), 348–375.
- A. R. Conn, K. Scheinberg, and Ph. L. Toint (1997a). On the convergence of derivative-free methods for unconstrained optimization. In A. Iserles and M. Buhmann, eds., *Approximation Theory and Optimization: Tributes to M. J. D. Powell*, pp. 83–108, Cambridge University Press, Cambridge, England.
- Summary.** Derivative-free trust-region methods for unconstrained optimization, inspired by Powell (1994b), are discussed and global convergence results obtained. The developments make extensive use of an interpolation error bound derived by Sauer and Xu (1995) in the context of multivariate polynomial interpolation.
- A. R. Conn, K. Scheinberg, and Ph. L. Toint (1997b). Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming, Series B*, **79**(3), 397–414.
- Summary.** Derivative-free trust-region methods for unconstrained optimization are introduced. Motivation is given, and past developments in the field reviewed. Techniques that ensure a suitable “geometric quality” of the models are considered. A discussion of open questions and perspectives is given.
- A. R. Conn, K. Scheinberg, and Ph. L. Toint (1998). A derivative free optimization algorithm in practice. Technical Report TR98/11, Department of Mathematics, University of Namur, Belgium.
- Summary.** An algorithm is presented for optimizing, at first without constraints, a nonlinear function whose first-order derivatives exist but are unavailable. It is based on approximating the objective function by a quadratic polynomial interpolation model and using this model within a trust-region framework. Some practical properties of the algorithm are studied, including how it can be extended to solve certain constrained optimization problems. Computational results are presented for analytical problems and for real-life problems from the aeronautical industry.
- A. R. Conn and J. W. Sinclair (1975). Quadratic programming via a non-differentiable penalty function. Technical Report CORR 75/15, Faculty of Mathematics, University of Waterloo, ON, Canada.
- A. R. Conn and Ph. L. Toint (1996). An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Giannessi, eds., *Nonlinear Optimization and Applications*, pp. 27–47, Plenum, New York.
- Summary.** The use of multivariate interpolation techniques is explored in the context of methods for unconstrained optimization that do not require derivatives. An algorithm is proposed that uses quadratic models in a trust-region framework. It requires few evaluations of

- the objective function and is relatively insensitive to noise in the objective function values. Its performance is analysed on a set of 20 examples, both with and without noise.
- A. R. Conn, L. N. Vicente, and C. Visweswarah (1999). Two-step algorithms for nonlinear optimization with structured applications. *SIAM Journal on Optimization*, **9**(4), 924–947.
- Summary.** Extensions to trust-region and linesearch algorithms, in which the classical step is augmented with a second step that yields a decrease of the objective function, are proposed. The convergence theory for trust-region and linesearch algorithms is adapted to this class of two-step algorithms. It is shown that the algorithms are globally convergent. The algorithms can be applied to any problem with variable(s) whose contribution to the objective function is a known functional form. In the nonlinear programming package **LANCELOT**, such an algorithm has been applied to update slack variables and variables introduced to solve min-max problems, leading to enhanced optimization efficiency. Numerical results are presented.
- L. B. Contesse (1980). Une caractérisation complète des minima locaux en programmation quadratique. *Numerische Mathematik*, **34**, 315–332.
- M. Contreras and R. A. Tapia (1993). Sizing the BFGS and DFP updates: Numerical study. *Journal of Optimization Theory and Applications*, **78**(1), 93–108.
- Summary.** A strategy is developed for selectively sizing the approximate Hessian matrix before it is updated in the Broyden–Fletcher–Goldfarb–Shanno (BFGS) and Davidon–Fletcher–Powell (DFP) trust-region methods for unconstrained optimization. The numerical results suggest that sizing should not only be restricted to the first iteration. The results also show that, without sufficient sizing, DFP is vastly inferior to BFGS, but, when selectively sized, DFP is competitive with BFGS.
- G. Corradi (1997). A trust-region algorithm for unconstrained optimization. *International Journal of Computer Mathematics*, **65**(1-2), 109–119.
- Summary.** A trust-region method is proposed that is based on approximation of  $f(\cdot)$  and  $f'(\cdot)$  of higher order. A convergence analysis for the method is presented. Numerical results are reported.
- J. E. Coster, N. Stander, and J. A. Snyman (1996). Trust region augmented Lagrangian methods with secant Hessian updating applied to structural optimization. In *Proceedings of the ASME Design Engineering Technical Conference and Computers in Engineering Conference, August 18–22, 1996, Irvine, California*.
- Summary.** A globally convergent augmented Lagrangian algorithm is formulated for the optimal sizing design of truss structures. The bound-constrained minimizations are performed using the **SBMIN** trust-region algorithm, and the Hessian of the augmented Lagrangian is approximated using partitioned secant updating. The performance of the algorithm is evaluated for different secant updates on standard explicit and truss sizing optimization problems. The results show the formulation to be superior to other implementations of augmented Lagrangian methods and that the method may approach the performance of the state-of-the-art sequential quadratic programming and spherical approximation methods. Of the secant updates, the symmetric rank-1 update is superior to the other updates including the Broyden–Fletcher–Goldfarb–Shanno scheme. Secant updating may be usefully applied in contexts where structural analysis and optimization are performed simultaneously, as in the simultaneous analysis and design method. In such cases the functions are partially separable.
- R. W. Cottle, J.-S. Pang, and R. E. Stone (1992). *The Linear Complementarity Problem*. Academic Press, London.

- R. Courant (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, **49**, 1–23.
- CPLEX 6.0 (1998). *High-Performance Linear, Integer and Quadratic Programming Software*. ILOG SA, Gentilly, France.
- C. W. Cryer (1982). *Numerical Functional Analysis*. Oxford University Press, Oxford, England.
- J. Cullum and R. A. Willoughby (1980). The Lanczos phenomenon—an interpretation based upon conjugate gradient optimization. *Linear Algebra and Its Applications*, **29**, 63–90.
- A. R. Curtis, M. J. D. Powell, and J. K. Reid (1974). On the estimation of sparse Jacobian matrices. *Journal of the Institute of Mathematics and Its Applications*, **13**, 117–119.
- S. Dafermos (1980). Traffic equilibrium and variational inequalities. *Transportation Science*, **14**, 42–54.
- J. W. Daniel (1967a). The conjugate gradient method for linear and nonlinear operator equations. *SIAM Journal on Numerical Analysis*, **4**, 10–25.
- J. W. Daniel (1967b). Convergence of the conjugate gradient method with computationally convenient modifications. *Numerische Mathematik*, **10**, 125–131.
- G. B. Dantzig (1963). *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, USA.
- I. Das (1996). An interior point algorithm for the general nonlinear programming problem with trust region globalization. Technical Report 96–61, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA.

**Summary.** A sequential quadratic programming (SQP)-based interior-point technique is presented for solving the general nonlinear programming problem using trust-region globalization and the Coleman and Li (1996a) scaling. The quadratic programming subproblem is decomposed into a normal and a reduced tangential subproblem, and strict feasibility is maintained with respect to the bounds. Computational experiments have been geared towards improving the semilocal convergence of the algorithm; in particular, high sensitivity of the speed of convergence with respect to the fraction of the trust-region radius allowed for the normal step and with respect to the initial trust-region radius are observed. The chief advantages of this algorithm over primal-dual interior-point algorithms are better handling of the “sticking problem” (that is, the difficulties that arise when the iterates approach, and are unable to leave, and incorrect active face) and a reduction in the number of variables by elimination of the multipliers of bound constraints.

- W. C. Davidon (1968). Variance algorithms for minimization. *Computer Journal*, **10**, 406–410.
- W. C. Davidon (1975). Optimally conditioned optimization algorithms without line searches. *Mathematical Programming*, **9**(1), 1–30.

C. De Boor and A. Ron (1992). Computational aspects of polynomial interpolation in several variables. *Mathematics of Computation*, **58**(198), 705–727.

T. De Luca, F. Facchinei, and Ch. Kanzow (1995). A semismooth approach to the solution of nonlinear complementarity problems. Technical Report 93, Institute of Applied Mathematics, University of Hamburg, Germany.

R. J. B. de Sampaio, J. Yuan, and W. Sun (1997). Trust region algorithm for nonsmooth optimization. *Applied Mathematics and Computation*, **85**(2–3), 109–116.

**Summary.** Minimization of a composite function  $h(f(x))$  is considered, where  $f : \mathbb{R}^n \leftarrow \mathbb{R}^m$  is a locally Lipschitzian function and  $h : \mathbb{R}^m \leftarrow \mathbb{R}$  is a continuously differentiable convex function. The theory of trust-region algorithms for nonsmooth optimization given by Fletcher (1987a) and Powell and Yuan (1990) is extended to this case. A trust-region algorithm and its global convergence are studied. Some applications to nonlinear and nonsmooth least-squares problems are given.

R. J. B. de Sampaio, J. Yuan, and W. Sun (1999). Quasi-Newton trust region algorithm for non-smooth least squares problems. *Applied Mathematics and Computation*, **105**(2–3), 183–194.

**Summary.** A quasi-Newton trust-region algorithm for nonsmooth least squares problems is proposed for the case where the functions may be decomposed into the sum of smooth and nonsmooth terms. This method uses a smooth subproblem with second-order information to approximate the locally Lipschitzian function, which both improves the convergence rate of the algorithm and copes with the possibility of large residuals. The global and superlinear convergence of the algorithm is established.

B. De Schutter and B. De Moor (1997). The extended linear complementarity problem and its applications in the max-plus algebra. In M. C. Ferris and J. S. Pang, eds., *Complementarity and Variational Problems: State of the Art*, pp. 22–39, SIAM, Philadelphia, USA.

E. J. Dean (1992). A model trust-region modification of Newton method for nonlinear 2-point boundary-value-problems. *Journal of Optimization Theory and Applications*, **75**(2), 297–312.

**Summary.** The method of quasilinearization for nonlinear two-point boundary value problems is Newton's method for a nonlinear differential operator equation. A trust-region approach to globalizing the quasilinearization algorithm is presented. A double-dogleg implementation yields a globally convergent algorithm that is robust in solving difficult problems.

R. S. Dembo, S. C. Eisenstat, and T. Steihaug (1982). Inexact-Newton methods. *SIAM Journal on Numerical Analysis*, **19**(2), 400–408.

R. S. Dembo and T. Steihaug (1983). Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, **26**, 190–212.

R. S. Dembo and U. Tulowitzki (1983). On the minimization of quadratic functions subject to box constraints. School of Organization and Management Working Paper Series B, No. 71, Yale University, New Haven, CT, USA.

V. F. Dem'yanov and V. N. Malozemov (1974). *Introduction to Minmax*. Wiley, Chichester, England.

- N. Deng, Y. Xiao, and F. Zhou (1993). Nonmonotonic trust region algorithms. *Journal of Optimization Theory and Applications*, **76**(2), 259–285.

**Summary.** A nonmonotonic trust-region method for unconstrained optimization is presented, whose convergence properties are similar to those for the usual trust-region method in which an approximate solution to the subproblem suffices. An algorithm to solve the subproblem is also given, and numerical results discussed.

- J. E. Dennis (1978). A brief introduction to quasi-Newton methods. In G. H. Golub and J. Oliker, eds., *Numerical Analysis*, Proceedings of Symposia in Applied Mathematics 22, pp. 19–52, American Mathematical Society, Providence, RI, USA.

- J. E. Dennis, N. Echebest, M. T. Guardarucci, J. M. Martínez, H. D. Scolnik, and C. Vacchino (1991). A curvilinear search using tridiagonal secant updates for unconstrained optimization. *SIAM Journal on Optimization*, **1**(3), 333–357.

- J. E. Dennis, M. El-Alem, and M. C. Maciel (1997). A global convergence theory for general trust-region based algorithms for equality constrained optimization. *SIAM Journal on Optimization*, **7**(1), 177–207.

**Summary.** A class of algorithms based on the sequential quadratic programming method for general nonlinear programming is presented. The objective function and the constraints of the problem are only required to be differentiable, and their gradients to satisfy a Lipschitz condition. Global convergence is obtained by using a trust-region approach together with an augmented Lagrangian merit function. A possibly nonmonotone updating technique is introduced for the penalty parameter. Global convergence results are proved and numerical experiments presented.

- J. E. Dennis, M. El-Alem, and K. A. Williamson (1999). A trust-region approach to nonlinear systems of equalities and inequalities. *SIAM Journal on Optimization*, **9**(2), 291–315.

**Summary.** Two one-sided trust-region algorithms for the numerical solution of systems of nonlinear equalities and inequalities are introduced. The first is a single-model algorithm, while the second uses multiple models with the Cauchy point computation being used as a model selection procedure. Global convergence analysis is presented for both algorithms, and numerical experiments show their effectiveness.

- J. E. Dennis, D. M. Gay, and R. E. Welsch (1981). An adaptive nonlinear least squares algorithm. *ACM Transactions on Mathematical Software*, **7**(3), 348–368.

- J. E. Dennis, M. Heinkenschloss, and L. N. Vicente (1998). Trust-region interior-point SQP algorithms for a class of nonlinear programming problems. *SIAM Journal on Control and Optimization*, **36**(5), 1750–1794.

**Summary.** Trust-region interior-point sequential quadratic programming algorithms are presented for solving minimization problems with nonlinear equality constraints and simple bounds. The algorithms treat states and controls as independent variables and take advantage of the structure of the problem. In particular, they do not rely on matrix factorizations of the linearized constraints but use solutions of the linearized state equation and the adjoint equation. They are suited for large-scale problems arising from optimal control problems governed by partial differential equations. They keep strict feasibility with respect to the bound constraints by using an affine-scaling method inspired by Coleman and Li (1996a), and they exploit trust-region techniques for equality constrained optimization. They allow the computation of the steps using a variety of methods, including many iterative techniques. Global

convergence is proved under very mild conditions on the trial steps. Under more stringent conditions on the quadratic model and on the trial steps, the iterates converge Q-quadratically to a limit point satisfying the second-order necessary conditions. Numerical results are reported for an optimal control problem governed by a nonlinear heat equation.

- J. E. Dennis, S. B. B. Li, and R. A. Tapia (1995). A unified approach to global convergence of trust region methods for nonsmooth optimization. *Mathematical Programming*, **68**(3), 319–346.

**Summary.** The global convergence of trust-region methods for nonsmooth minimization is investigated. Conditions are found on the local models that imply three convergence properties of such methods for regular problems. These conditions are satisfied by appropriate forms of Fletcher's (1987a) method for constrained optimization, Powell (1983) and Yuan's (1983) method for solving nonlinear fitting problems, and Duff, Nocedal, and Reid's (1987) and El-Hallabi and Tapia's (1993) methods for solving systems of nonlinear equations. The results may thus be viewed as a unified convergence theory for trust-region methods for nonsmooth problems.

- J. E. Dennis and H. H. W. Mei (1979). Two new unconstrained optimization algorithms which use function and gradient values. *Journal of Optimization Theory and Applications*, **28**(4), 453–482.

**Summary.** Two methods for unconstrained optimization are presented. They employ a hybrid direction strategy, which is a modification of Powell's (1970c) dogleg strategy, and a projection technique introduced by Davidon (1975) which uses projection images of  $\Delta x$  and  $\Delta g$  in updating the approximate Hessian. The first method uses Davidon's optimally conditioned update formula, while the second uses only the Broyden–Fletcher–Goldfarb–Shanno update. Both methods performed well without Powell's special iterations and singularity safeguards.

- J. E. Dennis and R. B. Schnabel (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted as Classics in Applied Mathematics 16, SIAM, Philadelphia, USA, 1996.

- J. E. Dennis and V. Torczon (1991). Direct search methods on parallel machines. *SIAM Journal on Optimization*, **1**(4), 448–474.

- J. E. Dennis and V. Torczon (1997). Managing approximation models in optimization. In N. M. Alexandrov and M. Y. Hussaini, eds., *Multidisciplinary Design Optimization*, pp. 330–347, SIAM, Philadelphia, USA.

**Summary.** It is standard engineering practice to use approximation models in place of expensive simulations to drive an optimal design process based on nonlinear programming algorithms. Known notions on trust-region methods and a global convergence theory for pattern search methods are used to manage the interplay between optimization and the fidelity of the approximation models to insure that the process converges to a reasonable solution of the original problem. The algorithm given as an example is based on the family of pattern search algorithms by Dennis and Torczon (1991), which can be successfully applied when only ranking information is available and when derivatives are either unavailable or unreliable.

- J. E. Dennis and L. N. Vicente (1996). Trust region interior-point algorithms for minimization problems with simple bounds. In H. Fisher, B. Riedmüller and S. Schäffler, eds., *Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter*, pp. 97–107, Physica-Verlag, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** Two trust-region interior-point algorithms for the solution of minimization problems with simple bounds are presented in which the local model is scaled as proposed by Coleman and Li (1996a). In the first, the trust region and the local quadratic model are consistently scaled. The second uses an unscaled trust region. A first-order convergence result is given, and dogleg and conjugate gradient algorithms to compute trial steps introduced. Numerical examples illustrate the advantages of the second algorithm.

- J. E. Dennis and L. N. Vicente (1997). On the convergence theory of trust-region based algorithms for equality-constrained optimization. *SIAM Journal on Optimization*, **7**(4), 927–950.

**Summary.** A global second-order convergence theory is given for the algorithm of Dennis, El-Alem, and Maciel (1997) that generalizes second-order convergence results of Moré and Sorensen (1983). The behaviours of the trust-region radius and the local rate of convergence are analysed. Some results concerning the trust-region subproblem for the linearized constraints, the quasi-normal component of the step, and the hard case are presented. It is shown how these results can be applied to some discretized optimal control problems.

- J. E. Dennis and K. A. Williamson (1988). A new parallel optimization algorithm for parameter identification in ordinary differential equations. In *Proceedings of the 27th IEEE Conference on Decision and Control*, Vol. 3, pp. 1836–1840, IEEE, New York, USA.

**Summary.** A variant of the Celis, Dennis, and Tapia (1985) trust-region algorithm for equality constrained optimization problems is described in the context of parameter identification in ordinary differential equations

- P. Deuflhard and F. A. Potra (1992). Asymptotic mesh independence for Newton–Galerkin methods via a refined Mysovskii theorem. *SIAM Journal on Numerical Analysis*, **29**(5), 1395–1412.

- S. Di and W. Sun (1996). A trust region method for conic model to solve unconstrained optimization. *Optimization Methods and Software*, **6**(4), 237–263.

**Summary.** A trust-region method using conic models is proposed for solving unconstrained optimization problems. Necessary and sufficient conditions for the solution of the associated subproblems are given. The method is globally and Q-superlinearly convergent. Numerical experiments are reported.

- G. Di Pillo, F. Facchinei, and L. Grippo (1992). An RQP algorithm using a differentiable exact penalty function for inequality constrained problems. *Mathematical Programming*, **55**(1), 49–68.

- G. Di Pillo and L. Grippo (1985). A continuously differentiable exact penalty-function method for nonlinear programming with inequality constraints. *SIAM Journal on Control and Optimization*, **23**(1), 72–84.

- G. Di Pillo and L. Grippo (1986). An exact penalty-function method with global convergence properties for nonlinear programming problems. *Mathematical Programming*, **36**(1), 1–18.

- I. I. Dikin (1967). Iterative solution of problems of linear and quadratic programming. *Doklady Akademii Nauk USSR*, **174**, 747–748.

I. I. Dikin and V. I. Zorkaltsev (1980). *Iterative Solutions of Mathematical Programming Problems*. Nauka, Novosibirsk, Russia.

M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, and S. A. Santos (1997). Comparing the numerical performance of two trust-region algorithms for large-scale bound-constrained minimization. *Investigación Operativa*, **7**, 23–54.

**Summary.** The numerical performance of the BOX-QUACAN and LANCELOT software packages are compared on an extensive set of problems. Conclusions are drawn about the classes of problems for which each package performs better.

M. A. Diniz-Ehrhardt, M. A. Gomes-Ruggiero, and S. A. Santos (1998). Numerical analysis of leaving-face parameters in bound-constrained quadratic minimization. Technical Report 52/98, Department of Applied Mathematics, IMECC-UNICAMP, Campinas, Brasil.

S. P. Dirkse and M. C. Ferris (1995). The PATH solver: A non-monotone stabilization scheme for mixed complementarity problems. *Optimization Methods and Software*, **5**(2), 123–156.

A. Djang (1979). Algorithmic equivalence in quadratic programming. Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, USA.

J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart (1979). *LINPACK Users' Guide*. SIAM, Philadelphia, USA.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst (1998). *Numerical Linear Algebra for High-Performance Computers*. SIAM, Philadelphia, USA.

Z. Dostál (1997). Box constrained quadratic programming with proportioning and projections. *SIAM Journal on Optimization*, **7**(3), 871–887.

V. Druskin, A. Greenbaum, and L. Knizhnerman (1998). Using nonorthogonal Lanczos vectors in the computation of matrix functions. *SIAM Journal on Scientific Computing*, **19**(1), 38–54.

I. S. Duff (1997). Sparse numerical linear algebra: Direct methods and preconditioning. In I. Duff and A. Watson, eds., *The State of the Art in Numerical Analysis*, pp. 27–62, Oxford University Press, Oxford, England.

I. S. Duff, A. M. Erisman, and J. K. Reid (1986). *Direct Methods for Sparse Matrices*. Oxford University Press, Oxford, England.

I. S. Duff, J. Nocedal, and J. K. Reid (1987). The use of linear programming for the solution of sparse sets of nonlinear equations. *SIAM Journal on Scientific and Statistical Computing*, **8**(2), 99–108.

**Summary.** A trust-region algorithm for solving sparse sets of nonlinear equations is proposed. It is based on minimizing the  $\ell_1$  norm of the linearized residual within an  $\ell_\infty$ -norm trust region, thereby permitting linear programming techniques to be applied. The algorithm has sparsity advantages over the Levenberg–Morrison–Marquardt algorithm.

- I. S. Duff and J. K. Reid (1983). The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, **9**(3), 302–325.
- I. S. Duff and J. K. Reid (1996). Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems. *ACM Transactions on Mathematical Software*, **22**(2), 227–257.
- I. S. Duff, J. K. Reid, N. Munksgaard, and H. B. Neilsen (1979). Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite. *Journal of the Institute of Mathematics and Its Applications*, **23**, 235–250.
- J. C. Dunn (1980). Newton’s method and the Goldstein step-length rule for constrained minimization problems. *SIAM Journal on Control and Optimization*, **6**, 659–674.
- J. C. Dunn (1987). On the convergence of projected gradient processes to singular critical points. *Journal of Optimization Theory and Applications*, **55**, 203–216.
- J. P. Dussault, J. A. Ferland, and B. Lemaire (1986). Convex quadratic programming with one constraint and bounded variables. *Mathematical Programming*, **36**(1), 90–104.
- J. G. Ecker and R. D. Niemi (1975). A dual method for quadratic programs with quadratic constraints. *SIAM Journal on Applied Mathematics*, **28**(3), 568–576.
- Summary.** A dual method is developed for minimizing a convex quadratic function of several variables subject to inequality constraints on the same type of function. The method solves a sequence of dual programs via a modified penalty function technique that does not eliminate the dual constraints but ensures that they will be active at optimality. A numerical example is included.
- J. Eckstein (1993). Nonlinear proximal point algorithms using Bregman functions with applications to convex programming. *Mathematics of Operations Research*, **18**(1), 202–226.
- O. Edlund (1997). Linear M-estimation with bounded variables. *BIT*, **37**(1), 13–23.
- Summary.** A subproblem in the trust-region algorithm for nonlinear M-estimation by Ekblom and Madsen (1989) is to find the restricted step, by calculating the M-estimator of the linearized model, subject to an  $\ell_2$ -norm bound on the variables. It is shown that this subproblem can be solved by applying Hebden (1973) iterations to the minimizer of the Lagrangian function. The method is compared with an augmented Lagrangian implementation.
- O. Edlund, H. Ekblom, and K. Madsen (1997). Algorithms for non-linear M-estimation. *Computational Statistics*, **12**(3), 373–383.
- Summary.** Algorithms for nonlinear M-estimation are presented. A trust-region approach is used, where a sequence of estimation problems for linearized models is solved. Numerical tests involving four estimators and ten nonlinear data-fitting problems are performed.
- L. Edsberg and P. A. Wedin (1995). Numerical tools for parameter-estimation in ODE systems. *Optimization Methods and Software*, **6**(3), 193–217.

**Summary.** The numerical problem of estimating unknown parameters in systems of ordinary differential equations from complete or incomplete data is treated. A numerical method for the optimization part is presented, based on the Gauss–Newton method with a trust-region approach to subspace minimization for the weighted nonlinear least-squares problem. The method is implemented in Matlab and several test problems from applications, giving nonstiff and stiff ordinary differential equation systems, are treated.

- H. Einarsson and K. Madsen (1998). Cutting planes and trust-regions for nondifferentiable optimization. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong.

**Summary.** A bundle trust-region method is proposed for the minimization of piecewise smooth functions. At each iteration  $f$  is approximated by a piecewise linear min-max function that models the set of generalized gradients in the neighbourhood of the current iterate. Two trust regions are used: an inner one in which the piecewise linear approximation to the objective function is found, and an outer one in which the trial step is calculated. Initially the two radii are equal, but the inner radius may be smaller if the iterate is close to the intersection between smooth pieces. The method is tested on convex and nonconvex test problems.

- S. C. Eisenstat and H. F. Walker (1994). Choosing the forcing terms in an inexact Newton method. Technical Report 6/94/75, Department of Mathematics and Statistics, Utah State University, Logan, UT, USA.

- H. Ekblom and K. Madsen (1989). Algorithms for non-linear Huber estimation. *BIT*, **29**(1), 60–76.

- M. El-Alem (1988). A global convergence theory for a class of trust region algorithms for constrained optimization. Technical Report TR88-5, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A variant of the Celis, Dennis, and Tapia (1985) trust-region algorithm for equality constrained optimization is given. An augmented Lagrangian merit function is used, and a scheme for updating the penalty parameter presented. Global and local convergence analyses are given showing that the algorithm reduces to the standard sequential quadratic programming algorithm in a neighborhood of the minimizer. The global convergence theory is sufficiently general that it holds for any algorithm that generates steps giving at least a fraction of the Cauchy decrease in the quadratic model of the constraints.

- M. El-Alem (1991). A global convergence theory for the Dennis-Celis-Tapia trust-region algorithm for constrained optimization. *SIAM Journal on Numerical Analysis*, **28**(1), 266–290.

**Summary.** A global convergence theory for a class of trust-region algorithms for equality constrained optimization is presented, which holds for any algorithm that generates steps giving at least a fraction of the Cauchy decrease in the quadratic model of the constraints, and which uses the augmented Lagrangian as a merit function. This theory is used to establish global convergence of the Celis, Dennis, and Tapia (1985) algorithm with a different scheme for updating the penalty parameter. The behaviour of the penalty parameter is also discussed.

- M. El-Alem (1995a). Global convergence without the assumption of linear independence for a trust-region algorithm for constrained optimization. *Journal of Optimization Theory and Applications*, **87**(3), 563–577.

**Summary.** A trust-region algorithm for solving the equality constrained optimization problem is presented. This algorithm uses the (Byrd and) Omojokun (1989) mechanism for computing

the trial steps but it differs in the way steps are evaluated. Global convergence is proved without assuming linear independence of the constraints' gradients.

- M. El-Alem (1995b). A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization. *SIAM Journal on Optimization*, **5**(2), 348–378.

**Summary.** A trust-region algorithm for nonlinear optimization subject to equality constraints is introduced. In computing the trial step, a projected Hessian technique converts the trust-region subproblem to one similar to that of the unconstrained case. To force global convergence, the augmented Lagrangian is employed as a merit function. An updating scheme that allows the penalty parameter to be decreased whenever it is warranted is proposed. It is shown that this algorithm is globally convergent and that the globalization strategy does not disrupt fast local convergence. This theory is sufficiently general that it holds for any algorithm that generates steps whose normal component gives at least a fraction of the Cauchy decrease in the quadratic model of the constraints and that uses Fletcher's (1970a) exact penalty function as a merit function.

- M. El-Alem (1996a). Convergence to a 2nd order point of a trust-region algorithm with nonmonotonic penalty parameter for constrained optimization. *Journal of Optimization Theory and Applications*, **91**(1), 61–79.

**Summary.** It is shown that a subsequence of iterates produced by the trust-region algorithm of El-Alem (1995b) converges to a point that satisfies both the first- and second-order necessary conditions.

- M. El-Alem (1996b). A strong global convergence result for Dennis, El-Alem, and Maciel's class of trust region algorithms. Technical Report TR96-15, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A global convergence theory for Dennis, El-Alem, and Maciel's (1997) class of algorithms is presented, which is analogous to Thomas's (1975) result for unconstrained optimization. In particular, every accumulation point of the sequence of iterates is a first-order stationary point. This result generalizes many current global convergence theories for trust-region algorithms for equality-constrained minimization that use an augmented Lagrangian merit function.

- M. El-Alem (1999). A global convergence theory for Dennis, El-Alem, and Maciel's class of trust-region-based algorithms for constrained optimization without assuming regularity. *SIAM Journal on Optimization*, **9**(4), 965–990.

**Summary.** A convergence theory is presented for a general class of trust-region algorithms for solving the smooth nonlinear programming problem with equality constraints. The results are proved under very mild conditions on the quasi-normal and tangential components of the trial steps. The Lagrange multiplier estimates and the Hessian estimates are assumed to be bounded. In addition, no regularity assumption, such as linear independence of the constraints' gradients, is made. The theory proves global convergence to one of four different types of Mayer-Bliss stationary points and holds for any algorithm that uses the augmented Lagrangian as a merit function, the El-Alem (1995b) scheme for updating the penalty parameter, and bounded multiplier and Hessian estimates.

- M. El-Alem and R. A. Tapia (1995). Numerical experience with a polyhedral-norm CDT trust-region algorithm. *Journal of Optimization Theory and Applications*, **85**(3), 575–591.

**Summary.** A modification of the Celis, Dennis, and Tapia (1985) (CDT) trust-region subproblem, which is obtained by replacing the  $l_2$  norm with a polyhedral norm, is studied. The polyhedral norm CDT subproblem can be solved using a standard quadratic programming code. Computational results that compare the performance of the polyhedral-norm CDT trust-region algorithm with the performance of existing codes are given.

- A. S. El-Bakry (1998). Convergence rate of primal-dual reciprocal barrier Newton interior-point methods. *Optimization Methods and Software*, **9**(1–3), 37–44.
- M. El-Hallabi (1987). A global convergence theory for arbitrary norm trust region methods for nonlinear equations. Technical Report TR87-5, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** The Levenberg–Morrison–Marquardt (LMM) algorithm for approximating zeros of the nonlinear system  $F(x) = 0$  is generalized to allow the use of arbitrary norms in the objective function and in the trust-region constraint. The algorithm, which is motivated by that of Duff, Nocedal, and Reid (1987), is globally convergent. Essentially the same properties apply for the general and for the LMM algorithm. In this analysis, the sequence generated is the couple  $(x_k, \Delta_k)$ , where  $x_k$  is the iterate and  $\Delta_k$  the trust-region radius. Since the successor  $(x_{k+1}, \Delta_{k+1})$  of  $(x_k, \Delta_k)$  is not unique the algorithm is modelled by a point-to-set map, and a convergence theorem due to Zangwill is applied. The algorithm locally reduces to Newton's method.

- M. El-Hallabi (1990). A global convergence theory for a class of trust-region methods for nonsmooth optimization. Technical Report TR90-10, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A class of trust-region algorithms is defined to find an approximate minimizer of the function  $f = h(F)$ , where  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is continuously differentiable and  $h : \mathbb{R}^m \rightarrow \mathbb{R}$  is regular locally Lipschitz. Algorithms from this class are globally convergent. The analysis is a generalization of that given by El-Hallabi and Tapia (1993). The algorithms are a natural generalization of those for smooth minimization to nonsmooth optimization.

- M. El-Hallabi (1993). An inexact minimization trust-region algorithm: Globalization of Newton's method. Technical Report TR93-43, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A trust-region algorithm generalizing that of El-Hallabi and Tapia (1993) is given in which the trust-region subproblem can be solved approximately. The algorithm considered is globally convergent and its rate of convergence is Q-superlinear. Quadratic convergence can be obtained by requiring sufficient accuracy in the solution of the local model.

- M. El-Hallabi (1999). Globally convergent multi-level inexact hybrid algorithm for equality constrained optimization. Technical Report RT11-98(revised), Département Informatique et Optimisation, Institut National des Postes et Télécommunications, Rabat, Morocco.

**Summary.** The combination of linesearch and trust-region techniques is investigated in the context of problems with equality constraints. Beneficial effects of internal doubling in this context are discussed, together with an alternative constraint qualification.

- M. El-Hallabi and R. A. Tapia (1993). A global convergence theory for arbitrary norm trust-region methods for nonlinear equations. Technical Report TR93-41, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** The Levenberg–Morrison–Marquardt algorithm for approximating zeros of the nonlinear system  $F(x) = 0$ , where  $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is continuously differentiable, is extended. Arbitrary norms can be used in place of the  $\ell_2$  norm for the trust-region constraint. The algorithm is globally convergent. This algorithm is motivated by the work of Duff, Nocedal, and Reid (1987). It locally reduces to Newton’s method.

- M. El-Hallabi and R. A. Tapia (1995). An inexact trust-region feasible-point algorithm for nonlinear systems of equalities and inequalities. Technical Report TR95-09, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A feasible-point trust-region algorithm for approximating solutions of the nonlinear system of equalities and inequalities  $F(x, y) = 0, y \geq 0$ , where  $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$  is continuously differentiable, is considered. By exploiting the convex structure of the local trust-region subproblem, a globally convergent inexact trust-region feasible-point algorithm is suggested for minimizing an arbitrary norm of the residual,  $\|F(x, y)\|_a$ , subject to non-negativity constraints. This algorithm uses a descent direction that is an inexact solution of the trust-region subproblem and then uses linesearch techniques to obtain an acceptable steplength. It is shown that, under weak hypotheses, any accumulation point of the iteration sequence is a constrained stationary point for  $f = \|F\|_a$  and that the sequence of constrained residuals converges to zero.

- C. Elster and A. Neumaier (1997). A method of trust region type for minimizing noisy functions. *Computing*, **58**(1), 31–46.

**Summary.** The optimization of noisy functions of a few variables is a commonly occurring problem in application areas such as finding the optimal choice of a few control parameters in chemical experiments. The traditional tool for the treatment of such problems is the method of Nelder and Mead (1965) (NM). An alternative method based on a trust-region approach (TR) is proposed and compared to NM. On a standard collection of test functions for unconstrained optimization by Moré, Garbow, and Hillstrom (1981), TR is substantially more robust than NM. If performance is measured by the number of function evaluations, TR is seen to be, on average, twice as fast as NM.

- E. Eskow and R. B. Schnabel (1991). Algorithm 695: Software for a new modified Cholesky factorization. *ACM Transactions on Mathematical Software*, **17**(3), 306–312.
- D. J. Evans (1968). The use of pre-conditioning in iterative methods for solving linear equations with positive definite matrices. *Journal of the Institute of Mathematics and Its Applications*, **4**, 295–314.
- F. Facchinei, A. Fischer, and Ch. Kanzow (1997). A semismooth Newton method for variational inequalities: The case of box constraints. In M. C. Ferris and J. S. Pang, eds., *Complementarity and Variational Problems: State of the Art*, pp. 76–90, SIAM, Philadelphia, USA.
- F. Facchinei, J. Júdice, and J. Soares (1998). An active set Newton algorithm for large-scale nonlinear programs with box constraints. *SIAM Journal on Optimization*, **8**(1), 158–186.

- F. Facchinei and Ch. Kanzow (1997). On unconstrained and constrained stationary points of the implicit Lagrangian. *Journal of Optimization Theory and Applications*, **92**(1), 99–115.
- F. Facchinei and S. Lucidi (1993). Nonmonotone bundle-type scheme for convex nonsmooth minimization. *Journal of Optimization Theory and Applications*, **76**(2), 241–257.
- F. Facchinei and J. Soares (1997). A new merit function for nonlinear complementarity problems and a related algorithm. *SIAM Journal on Optimization*, **7**(1), 225–247.
- Y. Fan, S. Sarkar, and L. S. Lasdon (1988). Experiments with successive quadratic programming algorithms. *Journal of Optimization Theory and Applications*, **56**(3), 359–383.
- Summary.** Important issues in sequential quadratic programming methods include the choice of either linesearch or trust-region strategies and the quadratic program formulation to be used and how the quadratic program is to be solved. The quadratic programs proposed by Fletcher and Powell are considered and a specialized reduced-gradient procedure discussed for solving them. The various options are compared on some well-known test problems.
- X. Fei and W. Huachen (1998). Integrated algorithm for bilevel nonsmooth optimization problems. *Journal of Shanghai Jiaotong University*, **32**(12), 115–119 (in Chinese).
- Summary.** A 1 leader– $N$  followers bilevel nonsmooth optimization problem is considered. A trust-region-based bundle in which appropriate generalized second derivatives are obtained using a Davidon–Fletcher–Powell-like formula is given, which combines the global convergence properties of the bundle method with the fast local convergence properties resulting from the use of approximate second derivatives.
- U. Felgenhauer (1997). Algorithmic stability analysis for certain trust region methods. In A. V. Fiacco, ed., *Mathematical Programming with Data Perturbations*, pp. 109–131, Lecture Notes in Pure and Applied Mathematics 195, Marcel Dekker, New York, Basel.
- Summary.** Quasi-Newton trust-region methods for unconstrained and bound-constrained optimization are proven to be robust with respect to errors in the gradient. Global convergence and active constraint identifications are proved under the assumption that this error is bounded by a multiple of the trust-region radius and that the model's Hessians are bounded and nonzero.
- G. Feng (1998). Trust-region method with simplicial decomposition for linearly constrained problems. Technical Report, December 17, Department of Applied Mathematics, Tongji University, Shanghai, China.
- Summary.** A trust-region method is presented for the solution of pseudoconvex optimization problems subject to linear constraints. The method uses restricted simplicial decomposition to produce successive simplices that are included in the feasible domain, and a trust-region method is then employed to minimize the objective function on those simplices.
- G. Feng (1999). Combination of trust region method and simplicial decomposition for linearly constrained problems. Technical Report, Department of Applied Mathematics, Tongji University, Shanghai, China.
- Summary.** A variant of the method developed in Feng (1998) is presented, where no restriction on the steplength is imposed on the master problem.

- M. C. Ferris, C. Kanzow, and T. S. Munson (1999). Feasible descent algorithms for mixed complementarity problems. *Mathematical Programming*, **86**(3), 475–497.
- M. C. Ferris and J. S. Pang (1997a). Engineering and economic applications of complementarity problems. *SIAM Review*, **39**(4), 669–713.
- M. C. Ferris and J. S. Pang, editors (1997b). *Complementarity and Variational Problems: State of the Art*. SIAM, Philadelphia, USA.
- M. C. Ferris and S. K. Zavriev (1996). The linear convergence of a successive linear programming algorithm. Mathematical Programming Technical Report MP-TR-96-12, Computer Sciences Department, University of Wisconsin, Madison, WI, USA.
- Summary.** A successive linear programming algorithm for solving constrained nonlinear optimization problems is presented that uses an Armijo procedure for updating a trust-region radius. Linear convergence of the method is proved by relating the solutions of the subproblems to standard trust-region and gradient projection subproblems and adapting an error bound analysis of Luo and Tseng (1993). Computational results are provided for polyhedrally constrained nonlinear programs.
- A. V. Fiacco (1976). Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, **10**(3), 287–311.
- A. V. Fiacco (1983). *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Mathematics in Science and Engineering 165. Academic Press, London.
- A. V. Fiacco and G. P. McCormick (1963). Programming under nonlinear constraints by unconstrained optimization: A primal-dual method. Technical Report RAC-TP-96, Research Analysis Corporation, McLean, VA, USA.
- A. V. Fiacco and G. P. McCormick (1964a). Computational algorithm for the sequential unconstrained minimization technique for nonlinear programming. *Management Science*, **10**(4), 601–617.
- A. V. Fiacco and G. P. McCormick (1964b). The sequential unconstrained minimization technique for nonlinear programming: A primal-dual method. *Management Science*, **10**(2), 360–366.
- A. V. Fiacco and G. P. McCormick (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, Chichester, England. Reprinted as Classics in Applied Mathematics 4, SIAM, Philadelphia, USA, 1990.
- A. Fischer (1992). A special Newton-type optimization method. *Optimization*, **24**(3–4), 269–284.
- A. Fischer (1995). An NCP-function and its use for the solution of complementarity problems. In D. Du, L. Qi, and R. Womersley, eds., *Recent Advances in Nonsmooth Optimization*, pp. 88–105, World Scientific Publishers, Singapore.

- R. Fletcher (1970a). A class of methods for nonlinear programming with termination and convergence properties. In J. Abadie, ed., *Integer and Nonlinear Programming*, pp. 157–175, North-Holland, Amsterdam, the Netherlands.
- R. Fletcher (1970b). An efficient, globally convergent, algorithm for unconstrained and linearly constrained optimization problems. Technical Report TP 431, AERE Harwell Laboratory, Harwell, Oxfordshire, England.

**Summary.** An algorithm is described for minimization of nonlinear functions, subject possibly to linear constraints on the variables. At each iteration, a quasi-Newton Powell-symmetric-Broyden quadratic approximation of the objective function is minimized over a region in which the approximation is valid. A strategy for deciding when this region should be expanded or contracted is given. Global convergence is proved and numerical tests show that the algorithm is efficient in the number of function evaluations.

- R. Fletcher (1970c). A new approach to variable metric algorithms. *Computer Journal*, **13**, 317–322.
- R. Fletcher (1971a). A general quadratic programming algorithm. *Journal of the Institute of Mathematics and Its Applications*, **7**, 76–91.
- R. Fletcher (1971b). A modified Marquardt subroutine for nonlinear least-squares. Technical Report AERE-R 6799, AERE Harwell Laboratory, Harwell, Oxfordshire, England.

**Summary.** A Fortran subroutine is described for minimizing a sum of squares of functions of many variables. Such problems arise in nonlinear data fitting and in the solution of nonlinear algebraic equations. The subroutine is based on an algorithm due to Marquardt (1963), but with modifications that improve the performance of the method, yet which require negligible extra computer time and storage.

- R. Fletcher (1973). An exact penalty function for nonlinear programming with inequalities. *Mathematical Programming*, **5**(2), 129–150.
- R. Fletcher (1976). Factorizing symmetric indefinite matrices. *Linear Algebra and Its Applications*, **14**, 257–272.
- R. Fletcher (1980). *Practical Methods of Optimization. Volume 1: Unconstrained Optimization*. Wiley, Chichester, England.
- R. Fletcher (1981). *Practical Methods of Optimization. Volume 2: Constrained Optimization*. Wiley, Chichester, England.
- R. Fletcher (1982a). A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Studies*, **17**, 67–76.

**Summary.** Composite functions  $\phi(x) = f(x) + h(c(x))$ , where  $f$  and  $c$  are smooth and  $h$  is convex, encompass many nondifferentiable optimization problems of interest. Making a linear approximation to  $c(x)$  while including second-order terms in a quadratic approximation to  $f(x)$  is used to determine a composite function  $\psi$  that approximates  $\phi(x)$ , and an algorithm is proposed in which  $\psi$  is minimized on each iteration. If the trust-region technique is incorporated into the algorithm, then global convergence can be proved. It is shown that the above approximations ensure that a second-order rate of convergence is achieved.

- R. Fletcher (1982b). Second-order corrections for non-differentiable optimization. In G. A. Watson, ed., *Proceedings Dundee 1981*, pp. 85–114, Lecture Notes in Mathematics, Springer-Verlag, Heidelberg, Berlin, New York.
- R. Fletcher (1985). An  $\ell_1$  penalty method for nonlinear constraints. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., *Numerical Optimization 1984*, pp. 26–40, SIAM, Philadelphia, USA.
- R. Fletcher (1987a). *Practical Methods of Optimization*, second ed. Wiley, Chichester, England.
- R. Fletcher (1987b). Recent developments in linear and quadratic programming. In A. Iserles and M. J. D. Powell, eds., *The State of the Art in Numerical Analysis*, pp. 213–243, Oxford University Press, Oxford, England.
- R. Fletcher (1995). An optimal positive definite update for sparse Hessian matrices. *SIAM Journal on Optimization*, **5**(1), 192–217.
- R. Fletcher, N. I. M. Gould, S. Leyffer, and Ph. L. Toint (1999). Global convergence of trust-region SQP-filter algorithms for nonlinear programming. Technical Report 99/03, Department of Mathematics, University of Namur, Belgium.
- Summary.** Global convergence to first-order critical points is proved for two trust-region sequential quadratic programming filter algorithms of the type introduced by Fletcher and Leyffer (1997). The algorithms allow for an approximate solution of the quadratic subproblem and incorporate the safeguarding tests described in Fletcher, Leyffer, and Toint (1998). The first algorithm decomposes the step into its normal and tangential components, while the second replaces this decomposition by a stronger condition on the associated model decrease.
- R. Fletcher and M. P. Jackson (1974). Minimization of a quadratic function of many variables subject only to lower and upper bounds. *Journal of the Institute of Mathematics and Its Applications*, **14**(2), 159–174.
- R. Fletcher and S. Leyffer (1997). Nonlinear programming without a penalty function. Numerical Analysis Report NA/171, Department of Mathematics, University of Dundee, Scotland.
- Summary.** A sequential quadratic programming (SQP) trust-region algorithm for nonlinear programming is considered that is globally convergent without the need to use a penalty function. Instead, the concept of a “filter” is introduced which allows a step to be accepted if it reduces either the objective function or the constraint violation function. Numerical tests on a wide range of test problems are very encouraging, and the new algorithm compares favourably with LANCELOT and an implementation of *Sl<sub>1</sub>QP*.
- R. Fletcher and S. Leyffer (1998). User manual for filterSQP. Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Scotland.
- R. Fletcher, S. Leyffer, and Ph. L. Toint (1998). On the global convergence of an SLP-filter algorithm. Technical Report 98/13, Department of Mathematics, University of Namur, Belgium.

- Summary.** A mechanism for proving global convergence in filter-type trust-region methods for nonlinear programming is described. The main interest is to demonstrate how global convergence can be induced without forcing sufficient descent in a penalty-type merit function. The technique of proof allows a wide range of specific algorithmic choices associated with updating the trust-region radius and with feasibility restoration.
- R. Fletcher and E. Sainz de la Maza (1989). Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, **43**(3), 235–256.
- Summary.** Methods are considered for solving nonlinear programming problems using an exact  $\ell_1$  penalty function. Linear programming-like subproblems incorporating a trust-region constraint are solved successively both to estimate the active set and to provide a foundation for proving global convergence. In one particular method, second-order information is represented by approximating the reduced Hessian matrix, and Coleman and Conn (1982b) steps are taken. A criterion for accepting these steps is given that enables the superlinear convergence properties of the Coleman–Conn method to be retained while preserving global convergence and avoiding the Maratos (1978) effect. The methods generalize to solve a wide range of composite nonsmooth optimization problems, and the theory is presented in this general setting. Numerical experiments on small test problems are described.
- R. Fletcher and G. A. Watson (1980). First and second order conditions for a class of nondifferentiable optimization problems. *Mathematical Programming*, **18**(3), 291–307.
- O. E. Flippo and B. Jansen (1996). Duality and sensitivity in nonconvex quadratic optimization over an ellipsoid. *European Journal of Operational Research*, **94**(1), 167–178.
- Summary.** A duality framework for the problem of optimizing a nonconvex quadratic function over an ellipsoid is described. Additional insight is obtained by observing that this nonconvex problem is in a sense equivalent to a convex problem of the same type, from which known necessary and sufficient conditions for optimality readily follow. Based on the duality results, some existing solution procedures are interpreted as in fact solving the dual. The duality relations provide a natural framework for sensitivity analysis.
- R. Fontecilla (1990). Inexact secant methods for nonlinear constrained optimization. *SIAM Journal on Numerical Analysis*, **27**(1), 154–165.
- R. Fontecilla, T. Steihaug, and R. A. Tapia (1987). A convergence theory for a class of quasi-Newton methods for constrained optimization. *SIAM Journal on Numerical Analysis*, **24**(5), 1133–1151.
- M. Fortin and R. Glowinski (1982). *Méthodes de Lagrangien augmenté*, Méthodes mathématiques de l'informatique 9. Dunod, Paris, France.
- R. Fourer and S. Mehrotra (1993). Solving symmetrical indefinite systems for an interior-point method for linear programming. *Mathematical Programming, Series A*, **62**(1), 15–39.
- P. A. Fox, A. D. Hall, and N. L. Schryer (1978). The PORT mathematical subroutine library. *ACM Transactions on Mathematical Software*, **4**(2), 104–126.

- C. Fraley (1989). Software performance on nonlinear least-squares problems. Technical Report CS-TR-89-1244, Department of Operations Research, Stanford University, Stanford, CA, USA.

**Summary.** Numerical results are presented for a large set of problems using software that is widely available and has undergone extensive testing. The algorithms implemented include Newton-based linesearch and trust-region methods for unconstrained optimization, as well as Gauss–Newton, Levenberg–Morrison–Marquardt, and special quasi-Newton methods for nonlinear least-squares. The original intention was to use the best available software to compare the underlying algorithms, to identify classes of problems for each method on which the performance is either very good or very poor, and to provide benchmarks for future work in nonlinear least-squares and unconstrained optimization. The variability in the results makes it impossible to meet either of the first two goals; however, the results are significant as a step toward explaining why these aims are so difficult to accomplish.

- A. Frangioni and G. Gallo (1999). A bundle type dual-ascent approach to linear multicommodity min-cost flow problems. *INFORMS Journal on Computing*, **11**(4), 370–393.

**Summary.** A cost decomposition approach to the linear multicommodity min-cost flow problem is presented, in which the mutual capacity constraints are dualized and the resulting Lagrangian dual is solved with a dual-ascent bundle algorithm. This approach is shown to outperform a number of rival methods, especially on problems where the number of commodities is large with respect to the size of the graph. The specialized bundle algorithm is characterized by a new heuristic for the trust-region parameter handling and embeds a specialized quadratic programming solver that allows the efficient implementation of strategies for reducing the number of active Lagrangian variables.

- M. Frank and P. Wolfe (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3**, 95–110.

- R. M. Freund (1991). Theoretical efficiency of a shifted-barrier-function algorithm for linear programming. *Linear Algebra and Its Applications*, **152**, 19–41.

- A. Friedlander and J. M. Martínez (1994). On the maximization of a concave quadratic function with box constraints. *SIAM Journal on Optimization*, **4**(1), 177–192.

- A. Friedlander, J. M. Martínez, and M. Raydan (1995). A new method for large-scale box constrained convex quadratic minimization problems. *Optimization Methods and Software*, **5**(1), 57–74.

- A. Friedlander, J. M. Martínez, and S. A. Santos (1994a). A new trust region algorithm for bound constrained minimization. *Applied Mathematics and Optimization*, **30**(3), 235–266.

**Summary.** A method for maximizing a concave quadratic function with bounds on the variables is described, which combines conjugate gradients with gradient projection techniques, as in Moré and Toraldo (1991). A strategy for the decision of leaving the current face is introduced that ensures finite convergence even for a singular Hessian and in the presence of dual degeneracy. Numerical experiments are presented.

- A. Friedlander, J. M. Martínez, and S. A. Santos (1994b). On the resolution of linearly constrained convex minimization problems. *SIAM Journal on Optimization*, **4**(2), 331–339.

K. R. Frisch (1954). Principles of linear programming—with particular reference to the double gradient form of the logarithmic potential method. Memorandum of October 18, University Institute for Economics, Oslo, Norway.

K. R. Frisch (1955). The logarithmic potential function for convex programming. Memorandum of May 13, University Institute for Economics, Oslo, Norway.

M. Fu, Z. Q. Luo, and Y. Ye (1996). Approximation algorithms for quadratic programming. Technical Report, Department of Management Science, University of Iowa, Ames, IA, USA.

**Summary.** The problem of approximating the global minimum of a general quadratic program (QP) with  $n$  variables subject to  $m$  ellipsoidal constraints is considered. For  $m = 1$ , it is shown that the  $\epsilon$ -minimizer, where error  $\epsilon \in (0, 1)$ , can be obtained in polynomial time, meaning that the number of arithmetic operations is a polynomial in  $n$ ,  $m$ , and  $\log(1/\epsilon)$ . For  $m \geq 2$ , a polynomial time  $(1 - \frac{1}{m^2})$ -approximation algorithm is presented, as well as a semidefinite programming relaxation for this problem. In addition, approximation algorithms for solving the QP under the box constraints and the assignment polytope constraints are given.

P. Fugger (1996). Trust region subproblems and comparison with quasi-Newton methods. Master's thesis, Technical University of Graz, Austria.

**Summary.** One possibility for solving the trust-region subproblem is to apply a parametric eigenvalue problem (Rendl's method). The choice of the parameter of Rendl's method is decisive for the number of iterations needed to obtain a minimizer. The computational efficiency of this method is compared with that of the Davidon–Fletcher–Powell (DFP) algorithm.

K. Fujisawa, M. Kojima, and K. Nakata (1997). Exploiting sparsity in primal-dual interior-point methods for semidefinite programming. *Mathematical Programming, Series B*, **79**(1–3), 235–253.

M. Fukushima (1986). A successive quadratic-programming algorithm with global and superlinear convergence properties. *Mathematical Programming*, **35**(3), 253–264.

M. Fukushima (1992). Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical Programming, Series A*, **53**(1), 99–110.

M. Fukushima (1996). Merit functions for variational inequality and complementarity problems. In G. Di Pillo and F. Giannessi, eds., *Nonlinear Optimization and Applications*, pp. 155–170, Plenum, New York.

M. Fukushima, M. Haddou, H. V. Nguyen, J. J. Strodiot, T. Sugimoto, and E. Yamakawa (1996). A parallel descent algorithm for convex programming. *Computational Optimization and Applications*, **5**(1), 5–37.

**Summary.** A parallel decomposition algorithm for solving a class of convex optimization problem is proposed that contains convex programming problems with a strongly convex objective function. The algorithm is a variant of the trust-region method applied to the Fenchel dual of the problem. Global convergence is proved and computational experience is reported on the connection machine model CM-5.

M. Fukushima and L. Qi, editors (1998). *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*. Kluwer Academic Publishers, Dordrecht, the Netherlands.

M. Fukushima and Y. Yamamoto (1986). A second-order algorithm for continuous-time nonlinear optimal control problems. *IEEE Transactions on Automatic Control*, **AC-31**(7), 673–676.

**Summary.** A second-order algorithm is presented for the solution of continuous-time nonlinear optimal control problems. The algorithm is a trust-region variant of Newton's method and solves at each iteration a linear-quadratic control problem with an additional constraint. The algorithm is globally convergent. A numerical example illustrates the method.

B. P. Furey (1993). A sequential quadratic programming-based algorithm for optimization of gas networks. *Automatica*, **29**(6), 1439–1450.

**Summary.** An algorithm for optimal control of the British Gas network of pipes and controllable units over periods of up to a day is described. The problem is large scale and highly nonlinear in both objective function and constraints. The method is based on sequential quadratic programming and takes account of the structure of the pipeflow equations by means of a reduced gradient technique, which eliminates most of the variables from the quadratic subproblems. The latter involve only simple bound constraints, which are handled efficiently by a conjugate gradient–active set algorithm. Trust-region techniques permit use of the exact Hessian, preserving sparsity. More general constraints are handled at an outer level by a truncated augmented Lagrangian method. Results are included for some realistic problems.

S. A. Gabriel and J. J. Moré (1995). Smoothing of mixed complementarity problems. Technical Report MCS-P541-00995, Argonne National Laboratory, Argonne, IL, USA.

S. A. Gabriel and J. S. Pang (1994). A trust region method for constrained nonsmooth equations. In W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., *Large Scale Optimization: State of the Art*, pp. 155–181, Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** The convergence of a trust-region method for solving a system of nonsmooth equations subject to linear constraints is considered. The method is based on the existence of an iteration function for the nonsmooth equations and involves the solution of a sequence of subproblems defined by this function. A particular realization of the method leads to an arbitrary-norm trust-region method. Applications of the latter method to the nonlinear complementarity and related problems are discussed. Global convergence of the method and its rate of convergence are established under certain regularity conditions similar to those used in the nonsmooth equations/sequential quadratic programming method and its generalization. Computational results are reported.

W. Gander (1978). On the linear least squares problem with a quadratic constraint. Ph.D. thesis, Computer Science Department, Stanford University, Stanford, CA, USA.

W. Gander (1981). Least squares with a quadratic constraint. *Numerische Mathematik*, **36**, 291–307.

**Summary.** Properties of the solutions of the linear least-squares problem with a quadratic constraint are given and a numerical application discussed. The paper summarizes Gander (1978).

- L. Gao (1998). Using Huber method to solve  $l_1$ -norm problem. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 263–272, Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** The nondifferentiable  $\ell_1$  estimation problem is replaced by a sequence of smooth minimization problems using the Huber function to approximate the absolute value. An algorithm is proposed that uses a trust-region method to solve each of the subproblems. Numerical comparisons are made with competing approaches.

- M. R. Garey and D. S. Johnson (1979). *Computers and Intractability*. W. H. Freeman, New York, San Francisco.

- E. Garfield (1990). The most cited papers of all time, SCI 1945–1988. Part 3. Another 100 from the citation classics hall of fame. *Current Contents*, **34**, 3–13.

- D. M. Gay (1981). Computing optimal locally constrained steps. *SIAM Journal on Scientific and Statistical Computing*, **2**, 186–197.

**Summary.** The solution of the trust-region subproblem with ellipsoidal norms is considered. Procedures to cope with the case where the Hessian of the model of the objective function may be indefinite are presented, paying particular attention to the hard case. The proposed step-computing algorithm provides an attractive way to deal with negative curvature. Implementations of the algorithm have proved very satisfactory in the nonlinear least-squares solver NL2SOL.

- D. M. Gay (1982). On the convergence in trust-region algorithms for unconstrained optimization. Computing Science Technical Report 104, Bell Laboratories, Murray Hill, NJ, USA.

**Summary.** Convergence tests in the context of trust-region algorithms for solving unconstrained optimization problems are discussed. A general theorem that supports a diagnostic test for possible convergence to a minimizer at which the Hessian of the objective function is singular is given. If the gradient of the objective function is continuous, the theorem assures that either the “singular convergence” test is satisfied infinitely often for any positive convergence tolerance, or else the lengths of the steps taken tend to zero; moreover, if the model Hessians are locally bounded, then any limit point of the iterates is a critical point. This information can be used in a suite of convergence tests that involve easily understood tolerances and that provide helpful diagnostics.

- D. M. Gay (1983). Algorithm 611: Subroutines for unconstrained minimization using a model/trust-region approach. *ACM Transactions on Mathematical Software*, **9**(4), 503–524.

**Summary.** Subroutines for the minimization of a smooth function are provided. These codes work with exact or finite-difference gradients and exact or secant approximations to Hessians, using the reverse-communication paradigm. An approximation to the trust-region subproblem is computed using the double-dogleg technique of Dennis and Mei (1979).

- D. M. Gay (1984). A trust region approach to linearly constrained optimization. In D. F. Griffiths, ed., *Numerical Analysis: Proceedings Dundee 1983*, pp. 72–105, Lecture Notes in Mathematics 1066, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** A class of trust-region algorithms for solving linearly constrained optimization problems is suggested. The algorithms use a “local” active-set strategy to select their steps. This strategy is such that degeneracy and zero Lagrange multipliers do not slow convergence

(to a first-order stationary point) and that no anti-zigzagging precautions are necessary—when there are zero Lagrange multipliers, convergence to a point failing to satisfy second-order necessary conditions remains possible. Specialization of the algorithms to the case of simple bounds on the variables are discussed, and preliminary computational experience reported.

- D. M. Gay, M. L. Overton, and M. H. Wright (1998). A primal-dual interior method for nonconvex nonlinear programming. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 31–56, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- A. George and J. W. H. Liu (1979). The design of a user interface for a sparse matrix package. *ACM Transactions on Mathematical Software*, **5**(2), 139–162.
- A. George and J. W. H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- M. Germain and Ph. L. Toint (2000). An iterative process for international negotiations on acid rain in northern Europe using a general convex formulation. *Environmental and Resource Economics*, **15**(3), 199–216.
- Summary.** A game theoretical approach to international negotiations on transboundary pollution is proposed that uses a discrete time formulation. The resulting sequences of states is shown to converge from a noncooperative situation to an international optimum in a finite number of stages. A financial transfer structure is presented that makes the desired sequence of states individually rational and strategically stable. The concepts are applied in a numerical simulation of the  $SO_2$  transboundary pollution problem related to acid rain in Northern Europe, using a trust-region method to calculate the optimum at each negotiation stage.
- E. M. Gertz (1999). Combination trust-region line-search methods for unconstrained optimization. Ph.D. thesis, Department of Mathematics, University of California, San Diego, CA, USA.
- Summary.** Three unconstrained minimization methods are presented that combine trust-region and linesearch techniques, in the sense that a linesearch is performed in the direction obtained from the solution of the trust-region subproblem. The linesearch is also used to control the trust-region radius. Strong global convergence to first- and second-order points is proved and the asymptotic convergence properties of the algorithms analysed. Numerical results are presented that include a quasi-Newton Broyden–Fletcher–Goldfarb–Shanno variant of the algorithms.
- J. Ch. Gilbert (1991). Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization. *Mathematical Programming*, **50**(1), 1–28.
- P. E. Gill, N. I. M. Gould, W. Murray, M. A. Saunders, and M. H. Wright (1984). A weighted Gram-Schmidt method for convex quadratic programming. *Mathematical Programming*, **30**(2), 176–195.
- P. E. Gill and W. Murray (1974). Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, **7**(3), 311–350.
- P. E. Gill and W. Murray (1976). Minimization subject to bounds on the variables. NPL Report NAC 72, National Physical Laboratory, London, England.

- P. E. Gill and W. Murray (1978). Numerically stable methods for quadratic programming. *Mathematical Programming*, **14**(3), 349–372.
- P. E. Gill, W. Murray, D. B. Poncelet, and M. A. Saunders (1992). Preconditioners for indefinite systems arising in optimization. *SIAM Journal on Matrix Analysis and Applications*, **13**(1), 292–311.
- P. E. Gill, W. Murray, M. A. Saunders, G. W. Stewart, and M. H. Wright (1985). Properties of a representation of a basis for the null space. *Mathematical Programming*, **33**(2), 172–186.
- P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright (1986). On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical Programming*, **36**(2), 183–209.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1983). Computing forward-difference intervals for numerical optimization. *SIAM Journal on Scientific and Statistical Computing*, **4**(2), 310–321.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1985). Some issues in implementing a sequential quadratic programming algorithm. *SIGNUM Newsletter*, **20**(2), 13–19.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1988). Shifted barrier methods for linear programming. Technical Report SOL88-9, Department of Operations Research, Stanford University, Stanford, CA, USA.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1990). A Schur-complement method for sparse quadratic programming. In M. G. Cox and S. J. Hammarling, eds., *Reliable Scientific Computation*, pp. 113–138, Oxford University Press, Oxford, England.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1991). Inertia-controlling methods for general quadratic programming. *SIAM Review*, **33**(1), 1–36.
- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1992). Some theoretical properties of an augmented Lagrangian merit function. In P. M. Pardalos, ed., *Advances in Optimizations and Parallel Computing*, pp. 127–143, Elsevier, Amsterdam.
- P. E. Gill, W. Murray, and M. H. Wright (1981). *Practical Optimization*. Academic Press, London.
- T. Glad and E. Polak (1979). A multiplier method with automatic limitation of penalty growth. *Mathematical Programming*, **17**(2), 140–155.
- M. X. Goemans (1997). Semidefinite programming in combinatorial optimization. *Mathematical Programming, Series B*, **79**(1–3), 143–161.

- M. X. Goemans and D. P. Williamson (1996). Improved approximation algorithms for the maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, **42**, 1115–1145.
- D. Goldfarb (1970). A family of variable metric methods derived by variational means. *Mathematics of Computation*, **24**, 23–26.
- D. Goldfarb (1980). The use of negative curvature in minimization algorithms. Technical Report TR80-412, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** Algorithms for minimizing a nonlinear function of many variables that make use of negative curvature are examined. These algorithms can all be viewed as modified versions of Newton's method and their merits and drawbacks are discussed to help identify new and more promising methods. The algorithms considered include ones that compute and search along nonascent directions of negative curvature and ones that search along curvilinear paths generated by these directions and descent directions. Versions of the Goldfeldt, Quandt, and Trotter (1966) or, equivalently, methods based upon a trust-region strategy and gradient path methods are also considered. When combined with the numerically stable Bunch and Parlett (1971) factorization of a symmetric indefinite matrix, the latter two approaches give rise to efficient and robust minimization methods that can take advantage of negative curvature.

- D. Goldfarb and A. Idnani (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, **27**(1), 1–33.
- D. Goldfarb and S. C. Liu (1993). An  $O(n^3 L)$  primal dual potential reduction algorithm for solving convex quadratic programs. *Mathematical Programming*, **61**(2), 161–170.
- D. Goldfarb, S. C. Liu, and S. Wang (1991). A logarithmic barrier function algorithm for quadratically constrained convex quadratic programs. *SIAM Journal on Optimization*, **1**(2), 252–267.
- D. Goldfarb and Ph. L. Toint (1984). Optimal estimation of Jacobian and Hessian matrices that arise in finite difference calculations. *Mathematics of Computation*, **43**(167), 69–88.
- D. Goldfarb and S. Wang (1993). Partial-update Newton methods for unary, factorable and partially separable optimization. *SIAM Journal on Optimization*, **3**(2), 383–397.
- S. M. Goldfeldt, R. E. Quandt, and H. F. Trotter (1966). Maximization by quadratic hill-climbing. *Econometrica*, **34**, 541–551.

**Summary.** A gradient method for maximizing general functions is discussed. After a brief discussion of various known gradient methods, the mathematical foundation is laid for the algorithm that rests on maximizing a quadratic approximation to the function on a suitably chosen spherical region. The method requires no assumptions about the concavity of the function to be maximized and automatically modifies the stepsize in light of the success of the quadratic approximation to the function. Practical problems of implementing the algorithm are discussed, and computational experience presented.

- A. A. Goldstein (1964). Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, **70**, 709–710.
- G. H. Golub and D. P. O’Leary (1989). Some history of the conjugate gradient and Lanczos methods. *SIAM Review*, **31**(1), 50–102.
- G. H. Golub and C. F. Van Loan (1989). *Matrix Computations*, second ed. Johns Hopkins University Press, Baltimore, MD, USA.
- G. H. Golub and U. von Matt (1991). Quadratically constrained least squares and quadratic problems. *Numerische Mathematik*, **59**, 561–580.
- F. A. M. Gomes, M. C. Maciel, and J. M. Martínez (1999). Nonlinear programming algorithms using trust regions and augmented Lagrangians with nonmonotone penalty parameters. *Mathematical Programming*, **84**(1), 161–200.
- Summary.** An algorithm based on the sequential quadratic programming method for solving the general nonlinear programming problem is presented. The objective function and the constraints of the problem are only required to be differentiable and their gradients to satisfy a Lipschitz condition. The strategy for obtaining global convergence is based on a trust region. The merit function is a type of augmented Lagrangian. An updating scheme is introduced for the penalty parameter, by means of which monotone increase is not necessary. Global convergence results are proved and numerical experiments are presented.
- C. C. Gonzaga (1991). An interior trust region method for linearly constrained optimization. *COAL Newsletter*, **19**, 55–66.
- Summary.** The link between the ellipsoids associated with interior-point methods scaling and trust regions is exposed.
- M. D. Gonzalez-Lima, R. A. Tapia, and F. A. Potra (1998). On effectively computing the analytic center of the solution set by primal-dual interior-point methods. *SIAM Journal on Optimization*, **8**(1), 1–25.
- V. Gopal and L. T. Biegler (1997). Nonsmooth dynamic simulation with linear programming based methods. *Computers and Chemical Engineering*, **21**(7), 675–689.
- Summary.** A previously developed linear programming (LP) method is refined with the addition of a descent strategy that combines linesearch with trust-region approaches. The LP method has the advantage of naturally dealing with additional profile bounds. A method for the treatment of discontinuities occurring in dynamic simulation problems is also presented.
- F. J. Gould and J. W. Tolle (1972). Geometry of optimality conditions and constraint qualifications. *Mathematical Programming*, **2**(1), 1–18.
- N. I. M. Gould (1985). On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem. *Mathematical Programming*, **32**(1), 90–99.
- N. I. M. Gould (1986). On the accurate determination of search directions for simple differentiable penalty functions. *IMA Journal of Numerical Analysis*, **6**, 357–372.
- N. I. M. Gould (1989). On the convergence of a sequential penalty function method for constrained minimization. *SIAM Journal on Numerical Analysis*, **26**(1), 107–128.

- N. I. M. Gould (1991). An algorithm for large-scale quadratic programming. *IMA Journal of Numerical Analysis*, **11**(3), 299–324.
- N. I. M. Gould (1999a). Iterative methods for ill-conditioned linear systems from optimization. In G. Di Pillo and F. Gianessi, eds., *Nonlinear Optimization and Applications* 2, pp. 123–142, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- N. I. M. Gould (1999b). On modified factorizations for large-scale linearly-constrained optimization. *SIAM Journal on Optimization*, **9**(4), 1041–1063.
- N. I. M. Gould, M. E. Hribar, and J. Nocedal (1998). On the solution of equality constrained quadratic problems arising in optimization. Technical Report RAL-TR-98-069, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England.
- N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint (1998). A linesearch algorithm with memory for unconstrained optimization. In R. De Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., *High Performance Algorithms and Software in Nonlinear Optimization*, pp. 207–223, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint (1999). Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, **9**(2), 504–525.
- Summary.** When the number of variables is large, the most widely used strategy for solving the trust-region subproblem is to trace the path of conjugate gradient iterates either to convergence or until it reaches the trust-region boundary. Means of continuing the process once the boundary has been encountered are investigated. It is observed that the trust-region problem within the currently generated Krylov subspace has a very special structure which enables it to be solved efficiently. The proposed strategy is compared with existing methods.
- N. I. M. Gould and J. Nocedal (1998). The modified absolute-value factorization norm for trust-region minimization. In R. De Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., *High Performance Algorithms and Software in Nonlinear Optimization*, pp. 225–241, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Summary.** A trust-region method for unconstrained minimization, using a trust-region norm based upon a modified absolute-value factorization of the model Hessian, is proposed. It is shown that the resulting trust-region subproblem may be solved using a single factorization. In the convex case, the method reduces to a backtracking Newton linesearch procedure. Numerical experience suggests that the approach is effective in the nonconvex case.
- N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint (2000a). Superlinear convergence of primal-dual interior-point algorithms for nonlinear programming. Technical Report (in preparation), Department of Mathematics, University of Namur, Belgium.
- Summary.** The local convergence properties of interior-point algorithms for nonlinear programming are analysed. Such methods include that of Conn, Gould, Orban, and Toint (2000). It is shown that the method asymptotically requires the solution of a single linear system per outer iteration and converges with a nearly Q-quadratic rate.

- N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint (2000b). On the practical dependency of a trust-region algorithm on its parameters. Technical Report (in preparation), Department of Mathematics, University of Namur, Belgium.

**Summary.** An extensive numerical study of the statistically best ranges for the trust-region algorithm parameters is described, whose conclusions differ from folklore preconceptions. The impact of preconditioning on parameter choice and performance is discussed.

- N. I. M. Gould and Ph. L. Toint (1999). A note on the second-order convergence of optimization algorithms using barrier functions. *Mathematical Programming*, **85**(2), 433–438.

- N. I. M. Gould and Ph. L. Toint (2000). SQP methods for large-scale nonlinear programming. In M. J. D. Powell and S. Scholtes, eds., *Proceedings of the IFIP TC7 Conference on System Modelling and Optimization, Cambridge, 1999* (to appear), Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** A comparison is proposed of a number of recent sequential quadratic programming methods for the solution of large-scale nonlinear programming problems. Both linesearch and trust-region approaches are considered, as are the implications of interior-point and quadratic programming methods.

- A. S. Gow, X. Z. Guo, D. L. Liu, and A. Lucia (1997). Simulation of refrigerant phase equilibria. *Industrial and Engineering Chemistry Research*, **36**(7), 2841–2848.

**Summary.** Vapor-liquid equilibria for refrigerant mixtures modelled by an equation of state are studied. Phase behaviour calculated by the Soave–Redlich–Kwong (SRK) equation with a single adjustable binary interaction parameter is compared with experimental data for binary refrigerant mixtures, two with a supercritical component and one that exhibits azeotropic behaviour. It is shown that the SRK equation gives an adequate description of the phase envelope for binary refrigerant systems. The complex domain trust-region methods of Lucia, Guo, and Wang (1993) and Lucia and Xu (1994) are applied to fixed vapor, isothermal flash model equations, with particular attention to root finding and root assignment at the equation of state (EOS) level of the calculations, and convergence in the retrograde and azeotropic regions of the phase diagram. Rules for assigning roots to the vapor and liquid phases in the case where all roots to the EOS are complex-valued yield correct results, even in retrograde regions. Convergence of the flash model equations is also studied. It is shown that the complex domain trust-region algorithms outperform Newton’s method in singular regions of the phase diagram. A variety of geometric figures are used to illustrate salient points.

- A. Greenbaum (1997). *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, USA.

- A. Greenbaum and Z. Strakoš (1992). Predicting the behaviour of finite precision Lanczos and conjugate gradient computations. *SIAM Journal on Matrix Analysis and Applications*, **13**(1), 121–137.

- J. Greenstadt (1967). On the relative efficiencies of gradient methods. *Mathematics of Computation*, **21**, 360–367.

- A. Griewank (1989). On automatic differentiation. In M. Iri and K. Tanabe, eds., *Mathematical Programming: Recent Developments and Applications*, pp. 83–108, Kluwer Academic Publishers, Dordrecht, the Netherlands.

- A. Griewank (1994). Computational differentiation and optimization. In J. R. Birge and K. G. Murty, eds., *Mathematical Programming: State of the Art 1994*, pp. 102–131, University of Michigan, Ann Arbor, MI, USA.
- A. Griewank and G. Corliss (1991). *Automatic Differentiation of Algorithms: Theory, Implementation and Application*. SIAM, Philadelphia, USA.
- A. Griewank and Ph. L. Toint (1982a). On the unconstrained optimization of partially separable functions. In M. J. D. Powell, ed., *Nonlinear Optimization 1981*, pp. 301–312, Academic Press, London.
- A. Griewank and Ph. L. Toint (1982b). Partitioned variable metric updates for large structured optimization problems. *Numerische Mathematik*, **39**, 119–137.
- R. E. Griffith and R. A. Stewart (1961). A nonlinear programming technique for the optimization of continuous processing systems. *Management Science*, **7**, 379–392.
- Summary.** A simple method for solving nonlinear programming problems is given. A numerical example, a model construction example, and a description of a particular existing computer system are included in order to clarify the mode of operation of the method.
- L. Grippo, F. Lampariello, and S. Lucidi (1986). A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23**(4), 707–716.
- L. Grippo, F. Lampariello, and S. Lucidi (1989). A truncated Newton method with nonmonotone line search for unconstrained optimization. *Journal of Optimization Theory and Applications*, **60**(3), 401–419.
- L. Grippo, F. Lampariello, and S. Lucidi (1991). A class of nonmonotone stabilization methods in unconstrained optimization. *Numerische Mathematik*, **59**, 779–805.
- A. Grothey and K. McKinnon (1998). A superlinearly convergent trust region bundle method. Technical Report MS 98-015, Department of Mathematics and Statistics, University of Edinburgh, Scotland.
- Summary.** Current bundle methods for the minimization of nonsmooth functions converge at a linear rate. A superlinearly convergent bundle method, using a trust region, is proposed for nonconvex problems. Numerical experience on a power-generation problem is reported.
- W. A. Gruver and E. W. Sachs (1980). *Algorithmic Methods in Optimal Control*. Pitman, Boston, USA.
- C. Gurwitz (1994). Local convergence of a two-piece update of a projected Hessian matrix. *SIAM Journal on Optimization*, **4**(3), 461–485.
- W. Hackbusch (1994). *Iterative Solution of Large Sparse Systems of Equations*. Springer Series in Applied Mathematical Sciences, Springer-Verlag, Heidelberg, Berlin, New York.
- W. W. Hager (1987). Dual techniques for constrained optimization. *Journal of Optimization Theory and Applications*, **55**, 37–71.

- W. W. Hager (1999a). Stabilized sequential quadratic programming. *Computational Optimization and Applications*, **12**(1–2), 253–273.
- W. W. Hager (1999b). Minimizing a quadratic over a sphere. Technical Report, University of Florida, Gainesville, FL, USA.
- C. Han, P. Pardalos, and Y. Ye (1992). On the solution of indefinite quadratic problems using an interior point method. *Informatica*, **3**, 474–496.
- Q. Han and J. Han (1999). Modified quasi-Newton method with collinear scaling for unconstrained optimization. Technical Report February, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China.

**Summary.** A trust-region method is proposed for unconstrained minimization, where the model is obtained by a conic quasi-Newton update. The trust region is defined not in the original space but in the space of collinearly scaled variables. Limited numerical experience illustrates the practical potential of the method.

- S. P. Han (1977). A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, **22**(3), 297–309.
- S. P. Han and O. L. Mangasarian (1979). Exact penalty functions in nonlinear programming. *Mathematical Programming*, **17**(3), 251–269.
- S. P. Han and O. L. Mangasarian (1983). A dual differentiable exact penalty-function. *Mathematical Programming*, **25**(3), 293–306.
- M. Hanke (1997). A regularizing Levenberg–Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse Problems*, **13**(1), 79–95.

**Summary.** A Levenberg–Morrison–Marquardt scheme for nonlinear inverse problems is considered, where the corresponding Lagrange parameter is chosen by an inexact Newton strategy. This scheme is suitable for ill-posed problems as long as the Taylor remainder is of second order in the interpolating metric between the range and domain topologies. Estimates of this type are established for ill-posed parameter identification problems arising in inverse groundwater hydrology. Both transient and steady-state data are investigated. The performance of the scheme is compared to a usual implementation on a two-dimensional engineering model problem.

- R. J. Hanson and F. T. Krogh (1992). A quadratic-tensor model algorithm for nonlinear least-squares problems with linear constraints. *ACM Transactions on Mathematical Software*, **18**(2), 115–133.

**Summary.** A new algorithm is presented for solving linearly constrained nonlinear least-squares and nonlinear equation problems, based on approximating the nonlinear functions using the quadratic-tensor model. The algorithm uses a box-shaped trust region. The objective function is allowed to increase at intermediate steps, as long as the predictor indicates that a new set of best values exists in the trust region. There is logic provided to retreat to the current best values, if necessary. The algorithm is effective for problems with linear constraints and dense Jacobian matrices and appears to be efficient in terms of function and Jacobian evaluations.

- P. T. Harker and J. S. Pang (1990). Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming, Series B*, **48**(2), 161–220.
- P. T. Harker and B. Xiao (1990). Newton's method for the nonlinear complementarity problem: A B-differentiable equation approach. *Mathematical Programming, Series B*, **48**(3), 339–358.
- Harwell Subroutine Library (2000). *A catalogue of subroutines (HSL 2000)*. AEA Harwell Laboratory, Harwell, Oxfordshire, England.
- G. He, B. Diao, and Z. Gao (1997). An SQP algorithm with nonmonotone line search for general nonlinear constrained optimization problem. *Journal of Computational Mathematics*, **15**(2), 179–192.
- M. D. Hebden (1973). An algorithm for minimization using exact second derivatives. Technical Report T.P. 515, AERE Harwell Laboratory, Harwell, Oxfordshire, England.
- Summary.** A method for the solution of the  $\ell_2$  trust-region subproblem is proposed so that at each iteration the solution of a number (roughly two) of systems of linear equations is required, instead of an eigenvalue calculation.
- M. Heinkenschloss (1993). Mesh independence for nonlinear least squares problems with norm constraints. *SIAM Journal on Optimization*, **3**(1), 81–117.
- M. Heinkenschloss (1994). On the solution of a two ball trust region subproblem. *Mathematical Programming*, **64**(3), 249–276.
- Summary.** The structure of a two-ball trust-region subproblem arising in nonlinear parameter identification problems is investigated, and a method for its solution is proposed. The method decomposes the subproblem and allows the application of efficient methods for the solution of the trust-region subproblems. The discussion of the structure focuses on the case where both constraints are active and on the treatment of the unconstrained problem.
- M. Heinkenschloss (1998). A trust-region method for norm constrained problems. *SIAM Journal on Numerical Analysis*, **35**(4), 1594–1620.
- Summary.** A trust-region method for the solution of nonlinear optimization problems with norm constraints is presented and analysed. Such problems often arise in parameter identification or nonlinear eigenvalue problems. The algorithms studied allow for inexact gradient information and the use of subspace methods for the approximate solution of subproblems. Characterizations and the descent properties of trust-region steps are given, criteria for the existence of successful iterations under inexact gradient information and under the use of subspace methods are established, and global convergence of the method is proven.
- M. Heinkenschloss, M. Ulbrich, and S. Ulbrich (1999). Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity. *Mathematical Programming*, **86**(3), 615–635.
- Summary.** A class of affine-scaling interior-point methods for bound-constrained optimization problems is introduced which are locally Q-superlinearly or Q-quadratically convergent, even without assuming strict complementarity. The methods are derived from that of Coleman

- and Li (1994) but use a different scaling matrix and a projection of the step to maintain strict feasibility. Simple examples are presented to illustrate the pitfalls of Coleman and Li's approach in the degenerate case and to demonstrate the performance of the fast converging modifications.
- M. Heinkenschloss and L. N. Vicente (1995). Analysis of inexact trust region interior-point SQP algorithms. Technical Report CRPC-TR95546, Center for Research on Parallel Computers, Houston, TX, USA.
- Summary.** Inexact trust-region interior-point (TRIP) sequential quadratic programming (SQP) algorithms for the solution of optimization problems with nonlinear equality constraints and simple bound constraints are analysed. The effect of the inexactness in the computation of derivative information on the convergence of TRIP SQP is analysed, and practical rules to control the size of the associated residuals are given. It is shown that if the size of the residuals is of the order of both the size of the constraints and the trust-region radius, then the TRIP SQP algorithms are globally first-order convergent. Numerical experiments with two optimal control problems governed by nonlinear partial differential equations are reported.
- M. Heinkenschloss and L. N. Vicente (1999). Analysis of inexact trust region SQP algorithms. Technical Report 99-15, Department of Mathematics, University of Coimbra, Portugal.
- Summary.** The global convergence behaviour of a class of composite-step trust-region sequential quadratic programming (SQP) methods that allow inexact problem information is studied. This inexact information can result from iterative linear system solves within the trust-region SQP method or from approximations of first-order derivatives. Accuracy requirements are based on feasibility and optimality of the iterates. In the absence of inexactness, the analysis reduces to that of Dennis, El-Alem, and Maciel (1997). If all iterates satisfy the equality constraints, then the results are related to the known convergence properties for trust-region methods with inexact gradient information in unconstrained optimization.
- J. Heinz and P. Spellucci (1994). A successful implementation of the Pantoja-Mayne SQP method. *Optimization Methods and Software*, **4**(1), 1–28.
- H.-P. Helfrich and D. Zwick (1995). Trust region algorithms for the nonlinear least distance problem. *Numerical Algorithms*, **9**(1–2), 171–179.
- Summary.** Trust-region algorithms are presented for the nonlinearly constrained least-distance problem. Global convergence is proved.
- H. P. Helfrich and D. Zwick (1996). A trust region algorithm for parametric curve and surface fitting. *Journal of Computational and Applied Mathematics*, **73**(1–2), 119–134.
- Summary.** An algorithm is presented for solving the problem of finding a parameter (shape) vector such that the associated manifold is a best least-squares fit to scattered data. For robustness, the algorithm uses a globally convergent trust-region approach. The support set for the manifold may be all of  $\mathbb{R}^d$  or a closed, convex subset. In particular, it may be chosen so that our theory is applicable to splines.
- T. Helgaker (1991). Transition-state optimizations by trust-region image minimization. *Chemical Physics Letters*, **182**(5), 503–510.
- Summary.** A method for optimizing transition states is presented. The method combines Smith's image function with trust-region minimization. Calculations on HCN and  $C_2H_6$  illustrate the usefulness of the method for ab-initio potential energy surfaces. It is found that second-order image optimizations of transition states are as fast as conventional minimizations.

- T. Helgaker and J. Almlöf (1988). Molecular wave functions and properties calculated using floating Gaussian orbitals. *Journal of Chemical Physics*, **89**(8), 4889–4902.

**Summary.** The calculation of molecular electronic wave functions and properties using floating Gaussian orbitals (i.e., orbitals whose positions are optimized in space) is described. The wave function is optimized using a second-order convergent trust-region method and molecular properties up to second order are calculated analytically. The method is applied to a series of small molecules at the Hartree–Fock level using four different floating basis sets. Properties calculated using the floating double-zeta basis set augmented with one set of polarization functions and one set of diffuse orbitals are close to the Hartree–Fock limit.

- M. R. Hestenes (1969). Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, **4**, 303–320.

- M. R. Hestenes (1980). *Conjugate Direction Methods in Optimization*. Springer-Verlag, Heidelberg, Berlin, New York.

- M. R. Hestenes and E. Stiefel (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, **49**, 409–436.

- D. J. Higham (1999). Trust region algorithms and timestep selection. *SIAM Journal on Numerical Analysis*, **37**(1), 194–210.

**Summary.** The close relation between unconstrained optimization and systems of ordinary differential equations (ODEs) with gradient structure is exploited. The convergence properties of a trust region, or Levenberg–Morrison–Marquardt, algorithm are examined. The algorithm may also be regarded as a linearized implicit Euler method with adaptive timestep for gradient ODEs. Global convergence is established and the rate of convergence is superlinear, but not quadratic. The precise form of superlinear convergence is exhibited. In the gradient ODE context this result contributes to the theory of “gradient stability”. It also introduces the notion of adapting the timestep in order to control the rate at which equilibrium is approached. A related timestepping algorithm for general ODEs guarantees fast superlinear local convergence to a stable equilibrium. This algorithm has many applications to ODEs and semidiscretized partial differential equations where a steady state solution is required.

- N. J. Higham (1987). A survey of condition number estimation for triangular matrices. *SIAM Review*, **29**(4), 575–596.

- N. J. Higham (1993). *Writing for the Mathematical Sciences*. SIAM, Philadelphia, USA.

- N. J. Higham (1995). Stability of the diagonal pivoting method with partial pivoting. Numerical Analysis Report No. 265, Manchester Centre for Computational Mathematics, Manchester, England.

- N. J. Higham (1996). *Accuracy and Stability of Numerical Algorithms*. SIAM, Philadelphia, USA.

- N. J. Higham and S. Cheng (1998). Modifying the inertia of matrices arising in optimization. *Linear Algebra and Its Applications*, **275–276**, 261–279.

- J.-B. Hiriart-Urruty and C. Lemaréchal (1993a). *Convex Analysis and Minimization Algorithms. Part 1: Fundamentals*. Springer-Verlag, Heidelberg, Berlin, New York.

J.-B. Hiriart-Urruty and C. Lemaréchal (1993b). *Convex Analysis and Minimization Algorithms. Part 2: Advanced Theory and Bundle Methods*. Springer-Verlag, Heidelberg, Berlin, New York.

A. Holstad (1999). Numerical solution of nonlinear equations in chemical speciation calculations. *Computational Geosciences*, **3**(3–4), 229–257.

**Summary.** Speciation calculations involve the computation of the concentration of each individual chemical species in a multicomponent-multiphase chemical system. The numerical problem is to solve a system of coupled linear and nonlinear equations subject to the constraint that all unknowns are positive. The performance and accuracy of a series of nonlinear equation solvers are evaluated: A quasi-Newton method with the global step determined by different line search and trust region algorithms, the conjugate gradient method with the global step determined by linesearch, and the solvers in the codes TENSOLVE, CONMIN, and LBFGS.

L. R. Huang and K. F. Ng (1994). 2nd-order necessary and sufficient conditions in nonsmooth optimization. *Mathematical Programming*, **66**(3), 379–402.

IMSL (1999). *Fortran Numerical Library*. Visual Numerics, Houston, TX, USA.

B. M. Irons (1970). A frontal solution program for finite-element analysis. *International Journal on Numerical Methods in Engineering*, **2**, 5–32.

A. N. Iusem (1991). On the dual convergence and the rate of primal convergence of Bregman's convex programming method. *SIAM Journal on Optimization*, **1**(3), 401–423.

M. Jagersand (1995). Visual servoing using trust region methods and estimation of the full coupled visual-motor Jacobian. In *Proceedings of the IASTED International Conference. Applications of Control and Robotics*, pp. 105–108, IASTED-ACTA Press, Anaheim, CA, USA.

**Summary.** An algorithm for visual servoing, capable of learning the robot kinematics and camera calibration, is presented. The approach differs from previous work in that a full coupled Jacobian is estimated online without prior models and a trust-region method is used, improving stability and convergence of the controller. Experimental results on the positioning accuracy and convergence of this controller, showing an up to 5-fold improvement in repeatability on PUMA 761 and 762 robots and successful estimation of the Jacobian for 3, 6, and 12 controlled degrees of freedom with highly nonlinear transfer functions, are presented.

M. Jagersand, O. Fuentes, and R. Nelson (1996). Acquiring visual-motor models for precision manipulation with robot hands. In B. Buxton and R. Cipolla, eds., *Computer Vision – ECCV'96. 4th European Conference on Computer Proceedings*, Vol. 2, pp. 603–612, Springer-Verlag, Berlin, New York.

**Summary.** Dextrous high degree of freedom (DOF) robotic hands provide versatile motions for fine manipulation of potentially very different objects. However, fine manipulation of an object grasped by a multifinger hand is much more complex than if the object is rigidly attached to a robot arm. Creating an accurate model is difficult if not impossible. Instead, a combination of two techniques is proposed: the use of an approximate estimated motor model, based on the grasp tetrahedron acquired when grasping an object, and the use of visual feedback to achieve accurate fine manipulation. A novel active vision-based algorithm for visual serving is presented that is capable of learning the manipulator kinematics and

camera calibration online while executing a manipulation task. The approach differs from previous work in that a full, coupled image Jacobian is estimated online without prior models and that a trust-region control method is used, improving stability and convergence. Extensive experimental evaluation of visual model acquisition and visual servoing in three, four, and six DOF is presented.

- M. Jagersand, O. Fuentes, and R. Nelson (1997). Experimental evaluation of uncalibrated visual servoing for precision manipulation. In *Proceedings 1997 IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 2874–2880, IEEE, New York, USA.

**Summary.** An experimental evaluation of adaptive and nonadaptive visual servoing in three, six, and twelve degrees of freedom is compared to traditional joint feedback control. The main results are that positioning of a six-axis PUMA 762 arm is up to five times more precise under visual control than under joint control and that positioning of a Utah/MIT dextrous hand is better under visual control than under joint control by a factor of 2, and a trust-region-based adaptive visual feedback controller is very robust.

- V. K. Jain, T. E. McClellan, and T. K. Sarkar (1986). Half-Fourier transform and application to radar signals. In *ICASSP 86 Proceedings. IEEE-IECEJ-ASJ International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, pp. 241–244, IEEE, New York, USA.

**Summary.** The half Fourier transform is discussed and its application to radar-return signals with specular components examined. This transform enables the desired part to be separated from the specular impulsive components. The problem is numerically solved by applying a trust-region algorithm to a nonlinear least-squares formulation.

- H. Jarausch and W. Mackens (1987). Solving large nonlinear systems of equations by an adaptive condensation process. *Numerische Mathematik*, **50**(6), 633–653.

**Summary.** An algorithm that efficiently solves large nonlinear systems of the form  $Au = F(u)$ ,  $u \in \mathbb{R}^n$ , whenever an (iterative) solver “ $A^{-1}$ ” for the symmetric positive definite matrix  $A$  is available and  $F'(u)$  is symmetric, is described. Such problems arise from the discretization of nonlinear elliptic partial differential equations. By means of an adaptive decomposition process the original system is split into a low-dimensional system and a remaining high-dimensional system, which can easily be solved by fixed point iteration. A Newton-type trust-region algorithm is chosen for the treatment of the small system. Convergence results typical for trust-region algorithms carry over to the full iteration process.

- F. Jarre (1991). On the convergence of the method of analytic centers when applied to convex quadratic programs. *Mathematical Programming*, **49**(3), 341–358.

- F. Jarre (1998). A QQP-minimization method for semidefinite and smooth nonconvex programs. Presentation at the Optimization '98 Conference, Coimbra, Portugal.

**Summary.** An interior-point approach is presented for problems where the entries of a positive semidefinite matrix  $X$  have to be optimally determined in the presence of nonconvex constraints on the entries of  $X$ . The method combines ideas of a predictor-corrector interior-point method, of the sequential quadratic programming method, and of trust-region methods. Some convergence results are given and very preliminary numerical experiments discussed.

- F. Jarre and M. A. Saunders (1995). A practical interior-point method for convex programming. *SIAM Journal on Optimization*, **5**(1), 149–171.

D. Jensen and R. Polyak (1994). On the convergence of a modified barrier method for convex programming. *IBM Journal of Research and Development*, **38**(3), 307–320.

D. Jensen, R. Polyak, and R. Schneur (1992). Numerical experience with modified barrier functions method for linear programming. Research Report RC 18415, T. J. Watson Research Center, Yorktown Heights, NY, USA.

J. J. A. Jensen and H. Agren (1986). A direct, restricted-step, second-order MC SCF program for large scale *ab initio* calculations. *Chemical Physics*, **104**(2), 229–250.

**Summary.** A general-purpose Multiple-Configuration Self-Consistent Field program with a direct, fully second-order, and step-restricted algorithm is presented. Step control is trust-region based. Convergence to the lowest state of a symmetry is guaranteed, and test calculations show reliable convergence for excited states.

X. S. Ji, W. Kritpiphat, A. Aboudheir, and P. Tontiwachwuthikul (1999). Mass transfer parameter estimation using optimization technique: Case study in CO<sub>2</sub> absorption with chemical reaction. *Canadian Journal of Chemical Engineering*, **77**(1), 69–73.

**Summary.** A new approach of applying an optimization technique to simultaneously determine a physical liquid-film mass transfer coefficient ( $k(L)(o)$ ) and effective interfacial area ( $a(v)$ ) from a pilot plant data is considered. The mass transfer mechanism of the CO<sub>2</sub>-NaOH system is modelled using the two-film theory to represent the behaviors of packed absorbers. The model presents an overall absorption rate (R-v) as a function of  $k(L)(o)$  and  $a(v)$ . The optimization algorithm used in this study follows a modified Levenberg–Morrison–Marquardt method with a trust-region approach. The R-v predictions from the model are in good agreement with the experimental data, with an average error of 6.5%.

H. Jiang, M. Fukushima, L. Qi, and D. Sun (1998). A trust region method for solving generalized complementarity problems. *SIAM Journal on Optimization*, **8**(1), 140–158.

**Summary.** Based on a semismooth equation reformulation using Fischer's function, a trust-region algorithm is proposed for solving the generalized complementarity problem. It uses a generalized Jacobian of the function involved in the semismooth equation and adopts the squared natural residual of the semismooth equation as a merit function. Global convergence and, under a nonsingularity assumption, a local Q-superlinear (or quadratic) rate of convergence are established. Calculation of a generalized Jacobian is discussed and numerical results presented.

H. Jiang and L. Qi (1996). Globally and superlinearly convergent trust-region algorithms for convex SC<sup>1</sup>-minimization problems and its application to stochastic programs. *Journal of Optimization Theory and Applications*, **90**(3), 649–669.

**Summary.** A globally and superlinearly convergent trust-region algorithm for solving SC<sup>1</sup> problems is presented. Numerical examples are given on the application of this algorithm to stochastic quadratic programs.

H. Jiang and L. Qi (1997). A new nonsmooth equations approach to nonlinear complementarity problems. *SIAM Journal on Control and Optimization*, **35**(1), 178–193.

K. Jittorntrum and M. R. Osborne (1980). A modified barrier function method with improved rate of convergence for degenerate problems. *Journal of the Australian Mathematical Society (Series B)*, **21**, 305–329.

- O. Jonsson and T. Larsson (1990). A note on step-size restrictions in approximation procedures for structural optimization. *Computers and Structures*, **37**(3), 259–263.

**Summary.** Different ways of restricting the stepsize in iterative approximation methods for structural optimization problems by using either trust-region constraints or a penalty on the distance from the approximation point. The first approach is commonly used. It is demonstrated how a penalty term can be used instead.

- J. J. Júdice and F. M. Pires (1989). Direct methods for convex quadratic programs subject to box constraints. *Investigación Operacional*, **9**, 23–56.

- S. Kaniel (1966). Estimates for some computational techniques in linear algebra. *Mathematics of Computation*, **20**(95), 369–378.

- L. Kantorovich (1948). Functional analysis and applied mathematics. *Uspehi Matematicheskikh Nauk*, **3**, 89–185.

- Ch. Kanzow, N. Yamashita, and M. Fukushima (1997). New NCP-functions and their properties. *Journal of Optimization Theory and Applications*, **94**(1), 115–135.

- Ch. Kanzow and M. Zupke (1998). Inexact trust-region methods for nonlinear complementarity problems. In M. Fukushima and L. Qi, eds., *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp. 211–235, Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** The nonlinear complementarity problem is reformulated as a nonsmooth system of equations by using a recently introduced NCP function. A trust-region-type method is then applied to the resulting system of equations that allows an inexact solution of the trust-region subproblem. The algorithm is well defined for a general nonlinear complementarity problem and has global and local convergence properties. Numerical results are discussed.

- S. E. Karisch, F. Rendl, and H. Wolkowicz (1994). Trust regions and relaxations for the quadratic assignment problem. In P. M. Pardalos and H. Wolkowicz, eds., *Quadratic Assignment and Related Problems. DIMACS Workshop*, pp. 199–219, American Mathematical Society, Providence, RI, USA.

**Summary.** General quadratic matrix minimization problems, with orthogonal constraints, arise in continuous relaxations for the (discrete) quadratic assignment problem (QAP). Currently, bounds for QAP are obtained by treating the quadratic and linear parts of the objective function of the relaxations separately. It is shown how to handle general objectives as one function. Comparisons are made with standard trust-region subproblems. Numerical results are obtained using a parametric eigenvalue technique.

- N. Karmarkar (1984). A new polynomial-time algorithm for linear programming. *Combinatorica*, **4**, 373–395.

- W. Karush (1939). Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, IL, USA.

- L. C. Kaufman (1999). Reduced storage, quasi-Newton trust region approaches to function optimization. *SIAM Journal on Optimization*, **10**(1), 56–69.

- Summary.** A limited-memory quasi-Newton Broyden–Fletcher–Goldfarb–Shanno algorithm for unconstrained optimization is described that uses a dogleg trust-region scheme. This technique uses products with both the approximate Hessian and its inverse.
- X. Ke and J. Han (1995a). A class of nonmonotone trust region algorithms for constrained optimizations. *Chinese Science Bulletin*, **40**(16), 1321–1324.
- Summary.** The constrained optimization problem of minimizing a continuously differentiable function over a closed convex set is considered. A class of globally convergent nonmonotone trust-region algorithms is proposed for this problem.
- X. Ke and J. Han (1995b). A nonmonotone trust region algorithm for equality constrained optimization. *Science in China Series A—Mathematics, Physics, Astronomy, and Technological Sciences*, **38**(6), 683–695.
- Summary.** A nonmonotone trust region algorithm is proposed for the minimization of smooth functions subject to nonlinear equality constraints. It handles feasibility like Celis, Dennis, and Tapia (1985) and allows nonmonotonicity in the augmented Lagrangian which is used as a merit function.
- X. Ke and J. Han (1996). A nonmonotone trust region algorithm for unconstrained nonsmooth optimization. *Chinese Science Bulletin*, **41**(3), 197–201.
- Summary.** A nonmonotone trust-region method is presented for the solution of nonsmooth unconstrained problems. This algorithm uses the concept of “iteration functions” of Qi and Sun (1994). Global convergence to a Dini stationary point is proved.
- X. Ke, G. Liu, and D. Xu (1996). A nonmonotone trust region algorithm for unconstrained nonsmooth optimization. *Chinese Science Bulletin*, **41**(3), 197–201.
- Summary.** A globally convergent trust-region algorithm is proposed for unconstrained minimization of locally Lipschitzian functions that generalizes the approach of Qi and Sun (1994) by allowing a nonmonotone sequence of objective function values.
- N. Kehtarnavaz, M. Z. Win, and N. Mullani (1987). Estimation of diastole to systole changes from cardiac PET images. In *Proceedings of the Ninth Annual Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 2, pp. 850–851, IEEE, New York, USA.
- Summary.** The changes in the myocardium thickness, left ventricle diameter, and tracer activity between diastole and systole are estimated from cardiac positron-emission-tomography (PET) images. A comparative study is carried out between the IMSL mathematical subroutine library and the trust-region (TR) parameter estimation algorithm. It is shown that the TR algorithm converges regardless of the initial parameter values chosen. To reduce the number of iterations, a preprocessor has been developed to provide better starting values. The method is used as a diagnostic tool for abnormal heart conditions.
- E. L. Keller (1973). The general quadratic programming problem. *Mathematical Programming*, **5**(3), 311–337.
- C. T. Kelley and D. E. Keyes (1998). Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis*, **35**(2), 508–523.
- Summary.** Pseudotransient continuation is a well-known and physically motivated technique for computation of steady-state solutions of time-dependent partial differential equations. Standard globalization strategies such as linesearch or trust-region methods often stagnate

at local minima. Pseudotransient continuation succeeds in many of these cases by taking advantage of the underlying partial differential equation structure of the problem. Convergence for a generic form of pseudotransient continuation is proved and illustrated with two practical strategies.

- C. T. Kelley and E. W. Sachs (1987). Quasi-Newton methods and unconstrained optimal control problems. *SIAM Journal on Control and Optimization*, **25**(6), 1503–1517.

- C. T. Kelley and E. W. Sachs (1999). A trust region method for parabolic boundary control problems. *SIAM Journal on Optimization*, **9**(4), 1064–1081.

**Summary.** A trust-region algorithm for constrained parabolic boundary control problems is developed. The method is a projected form of the Steihaug–Toint method with a smoothing step added at each iteration to improve performance in the global phase and provide mesh-independent sup-norm convergence in the terminal phase.

- H. F. Khalfan, R. H. Byrd, and R. B. Schnabel (1999). Retaining convergence properties of trust region methods without extra gradient evaluations. Technical Report, Department of Mathematics and Computer Science, University of Colorado, Boulder, CO, USA.

**Summary.** A modification of a trust-region method based on the Powell-symmetric-Broyden quasi-Newton update is proposed that uses only the function value at rejected points to make the update at those points. The same  $q$ -superlinear convergence result as for the original algorithm holds for the new method.

- K. C. Kiwiel (1989a). An ellipsoid trust region bundle method for nonsmooth convex minimization. *SIAM Journal on Control and Optimization*, **27**(4), 737–757.

**Summary.** A bundle method of descent for minimizing a convex (possibly nonsmooth) function of several variables is presented. At each iteration the algorithm finds a trial point by minimizing a polyhedral model subject to an ellipsoid trust-region constraint. The quadratic matrix of the constraint, which is updated as in the ellipsoid method, is interpreted as a generalized “Hessian” to account for “second-order” effects, thus enabling faster convergence. Global convergence of the method is established and numerical results are given.

- K. C. Kiwiel (1989b). A survey of bundle methods for non-differentiable optimization. In M. Iri and K. Tanabe, eds., *Mathematical Programming: Recent Developments and Applications*, pp. 263–282, Kluwer Academic Publishers, Dordrecht, the Netherlands.

- K. C. Kiwiel (1996). Restricted step and Levenberg–Marquardt techniques in proximal bundle methods for nonconvex nondifferentiable optimization. *SIAM Journal on Optimization*, **6**(1), 227–249.

**Summary.** Two methods are given for minimizing locally Lipschitzian upper semidifferentiable functions. They employ extensions of restricted step (trust-region) and Levenberg–Morrison–Marquardt techniques. Extensions to linearly constrained optimization are discussed. Preliminary numerical experience is reported.

- O. Knoth (1983). Marquardt-ähnliche Verfahren zur Minimierung nichtlinearer Funktionen. Ph.D. thesis, Martin-Luther University, Halle-Wittenberg, Germany (in German).

- Summary.** Accumulation points of the trust-region Newton method for unconstrained optimization are proved to satisfy second-order optimality conditions. The use of negative curvature in a “projected” Marquardt algorithm is introduced. The characterization of local minima for the  $\ell_2$  trust-region subproblem is considered together with the use of an exact penalty function for this subproblem.
- D. E. Knuth (1973). *The Art of Computer Programming, Volume 3, Sorting and Searching*. Addison-Wesley, Reading, MA, USA.
- D. E. Knuth (1976). Big Omicron and Big Omega and Big Theta. *ACM SIGACT News*, **8**(2), 18–24.
- F. Kojima (1993). Back-propagation learning using the trust region algorithm and application to nondestructive testing in applied electromagnetics. *International Journal of Applied Electromagnetics in Materials*, **4**(7), 27–33.
- Summary.** An artificial neural network is applied by nondestructive inspections in aerospace materials. The back-propagation learning for a multilayer feed-forward neural network is applied to the resulting classification problem. The trust-region method is adapted to the back-propagation learning problem. Numerical tests are discussed.
- F. Kojima and H. Kawaguchi (1993). Backpropagation learning algorithm for nondestructive testing by thermal imager (aerospace materials). In *IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on Neural Networks*, Vol. 1, pp. 955–958, IEEE, New York, USA.
- Summary.** An artificial neural network is applied to nondestructive inspections using a thermal imager. The use of an artificial neural network is presented for classifying test data as corresponding to bonded and disbonded regions in sample materials. The back-propagation learning for a multilayer feed-forward neural network is applied to this classification. The trust-region method is adopted to the back-propagation learning problem. Numerical results are summarized.
- S. Kruk and H. Wolkowicz (1998). SQ<sup>2</sup>P, sequential quadratic constrained quadratic programming. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 177–204, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Summary.** An algorithm is proposed for constrained nonlinear programming, in which a quadratic model of the objective function is minimized, at each iteration, subject to quadratic approximations of the constraints (Q<sup>2</sup>P) and an additional trust region. As this subproblem is in general intractable, the Lagrangian relaxation of Q<sup>2</sup>P is instead solved using semidefinite programming. An example illustrates the advantages over the standard sequential quadratic programming approach.
- H. W. Kuhn and A. W. Tucker (1951). Nonlinear programming. In J. Neyman, ed., *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of Berkeley Press, Berkeley, CA, USA.
- H. H. Kwok, M. P. Kamat, and L. T. Watson (1985). Location of stable and unstable equilibrium-configurations using a model trust region quasi-Newton method and tunnelling. *Computers and Structures*, **21**(5), 909–916.
- Summary.** The combination of a quasi-Newton method with a deflation technique is proposed as an alternative to the hybrid method for locating multipole equilibrium configurations. The proposed method not only exploits sparsity and symmetry, but also represents an improvement in efficiency. A double-dogleg globalization strategy is used.

- M. Lalee, J. Nocedal, and T. D. Plantenga (1998). On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, **8**(3), 682–706.

**Summary.** A software implementation of (Byrd and) Omojokun's (1989) trust-region algorithm for nonlinear equality constrained optimization is described. The code is designed for the efficient solution of large problems and provides the user with a variety of linear algebra techniques for solving the subproblems occurring in the algorithm. Second derivative information can be used, as well as limited-memory quasi-Newton approximations. The performance of the code is studied using a set of difficult test problems from the CUTE collection.

- P. Lancaster and M. Tismenetsky (1985). *The Theory of Matrices*, second ed. Academic Press, London.

- C. Lanczos (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards B*, **45**, 225–280.

- A. Lannes (1997). Phase-closure imaging in algebraic graph theory: A new class of phase-calibration algorithms. *Journal of the Optical Society of America A. Optics, Image, Science, and Vision*, **15**(2), 419–429.

**Summary.** A class of phase-calibration algorithms is presented. Algebraic graph theory is used to exploit the particular structures present in such problems. The corresponding trust-region optimization process proves to be well suited to these structures. The main result is that the traditional notions of phase closure imaging can be understood and refined in a wider framework. This has implications in all the fields where the notion of phase closure plays a key role, such as weak-phase imaging in optical interferometry, radio imaging, and remote sensing by aperture synthesis.

- A. Lannes (1998). Weak-phase imaging in optical interferometry. *Journal of the Optical Society of America A. Optics, Image, Science, and Vision*, **15**(4), 811–824.

**Summary.** The first imaging devices of optical interferometry are typically likely to be of weak phase, that is, a set of three-element arrays independently observing the same object. The study of their imaging capabilities refers to appropriate optimization methods, which essentially address the self-calibration process and its stability. A general survey of these techniques is given, and it is shown, in particular, how the related algorithms can be used for examining the imaging capabilities of weak-phase interferometric devices. The phase-calibration algorithm involved in the self-calibration cycles is of trust-region type and uses algebraic graph theory.

- L. S. Lasdon, J. Plummer, and G. Yu (1995). Primal-dual and primal interior point algorithms for general nonlinear programs. *ORSA Journal on Computing*, **7**(3), 321–332.

**Summary.** An interior-point algorithm for general nonlinear programs is presented. Inequality constraints are converted to equalities with slack variables. All bounds are handled with a barrier term in the objective. The Karush–Kuhn–Tucker system of the resulting equality constrained barrier problem is solved directly by Newton's method. Primal-dual, primal, and primal-dual with trust-region variants are developed and evaluated. An implementation that utilizes the true Hessian of the Lagrangian and exploits Jacobian and Hessian sparsity is described. Computational results are presented and discussed.

- C. L. Lawson and R. J. Hanson (1974). *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted as Classics in Applied Mathematics 15, SIAM, Philadelphia, USA, 1995.
- P. Le Tallec and R. Glowinski (1989). *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia, USA.
- H. A. Le Thi (1997). Contribution à l'optimisation non convexe et l'optimisation globale. Habilitation thesis, University of Rouen, France.
- H. A. Le Thi (2000). An efficient algorithm for globally minimizing a quadratic function under convex quadratic constraints. *Mathematical Programming*, to appear.
- Summary.** A combination of an algorithm for the difference of convex functions and branch-and-bound is developed to find the global minimum of a quadratic function subject to a finite set of convex quadratic constraints. The trust-region subproblem is used as the main subproblem of the new algorithm. Numerical experiments illustrate the proposal.
- F. Leibfritz and E. W. Sachs (1999). Optimal static output feedback design using a trust region interior point method. Presentation at the First Workshop on Nonlinear Optimization “Interior-Point and Filter Methods”, Coimbra, Portugal.
- Summary.** The problem of designing feedback control laws is considered when a complete set of state variables is not available. The resulting nonlinear and nonconvex matrix optimization problem including semidefiniteness constraints for determining the optimal feedback gain is solved by a trust-region interior-point approach. Test examples from optimal output feedback design numerically demonstrate the usefulness of the approach.
- C. Lemaréchal and J. Zowe (1994). A condensed introduction to bundle methods in nonsmooth optimizations. In E. Spedicato, ed., *Algorithms for Continuous Optimization: The State of the Art*, pp. 357–382, NATO ASI Series C: Mathematical and Physical Sciences 434, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- M. Lescrenier (1991). Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM Journal on Numerical Analysis*, **28**(2), 476–495.
- Summary.** The behaviour of the trust-region algorithms of Conn, Gould, and Toint (1988a) for optimization with simple bounds is analysed in the case where the strict complementarity condition is violated. It is proved that inexact Newton methods lead to superlinear or quadratic rates of convergence, even if the set of active bounds at the solution is not completely identified. Practical criteria for stopping the inner iterations of the algorithms are deduced.
- K. Levenberg (1944). A method for the solution of certain problems in least squares. *Quarterly Journal on Applied Mathematics*, **2**, 164–168.
- Summary.** The standard method for solving least-squares problems which leads to nonlinear normal equations depends on a reduction of the residuals to linear form by first-order Taylor approximations taken about a trial solution for the parameters. If the usual least-squares procedure with these linear approximations yields new values for the parameters which are not sufficiently close to the trial values, the neglect of second- and higher order terms may invalidate the process. This difficulty may be alleviated by limiting the absolute values of the parameters and simultaneously minimizing the sum of squares of the approximating residuals under these “damped” conditions.

- E. S. Levitin and B. T. Polyak (1966). Constrained minimization problems. *USSR Computing Mathematics and Mathematical Physics*, **6**, 1–50.
- R. M. Lewis (1996). A trust region framework for managing approximation models in engineering optimization. AIAA Paper 96-4101, presented at the Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design, Bellevue, WA, USA.
- Summary.** Nonquadratic models are proposed for the trust-region minimization of expensive functions from engineering applications.
- W. Li (1996). Differentiable piecewise quadratic exact penalty functions for quadratic programs with simple bound constraints. *SIAM Journal on Optimization*, **6**(2), 299–315.
- W. Li (1997). A merit function and a Newton-type method for symmetric linear complementarity problems. In M. C. Ferris and J. S. Pang, eds., *Complementarity and Variational Problems: State of the Art*, pp. 181–203, SIAM, Philadelphia, USA.
- W. Li and J. Swetits (1993). A Newton method for convex regression, data smoothing, and quadratic programming with bounded constraints. *SIAM Journal on Optimization*, **3**(3), 466–488.
- Y. Li (1993). Centering, trust region, reflective techniques for nonlinear minimization subject to bounds. Technical Report TR93-1385, Department of Computer Science, Cornell University, Ithaca, NY, USA.
- Summary.** Motivation is provided for techniques that are important for the method of Coleman and Li (1996a). Numerical experience on some medium-size problems is included.
- Y. Li (1994a). On global convergence of a trust-region and affine scaling method for nonlinearly constrained minimization. Technical Report TR94-1462, Department of Computer Science, Cornell University, Ithaca, NY, USA.
- Summary.** Global convergence properties of the method by Li (1994b) are presented.
- Y. Li (1994b). A trust-region and affine scaling method for nonlinearly constrained minimization. Technical Report TR94-1463, Department of Computer Science, Cornell University, Ithaca, NY, USA.
- Summary.** A trust-region approach based on an  $\ell_2$ -norm subproblem is proposed for solving a nonlinear  $\ell_1$  problem. The (quadratic) approximation and the trust-region subproblem are defined using affine-scaling techniques. Explicit sufficient decrease conditions based on the approximations are suggested for obtaining a limit point satisfying complementarity, Kuhn–Tucker conditions, and the second-order necessary conditions.
- X. Liang and C. Xu (1997). A trust region algorithm for bound constrained minimization. *Optimization*, **41**(3), 279–289.
- Summary.** A trust-region algorithm is proposed for box constrained nonlinear optimization. At each step of the algorithm a quadratic model problem is minimized in a box. The global and quadratic convergence rates to a strong local minimizer are given. Computational results are presented to show the efficiency of the algorithm.

- A. Liao (1995). Solving unconstrained discrete-time optimal-control-problems using trust method. Technical Report CTC95TR230, Advanced Computing Research Institute, Cornell Theory Center, Ithaca, NY, USA.

**Summary.** A trust-region method is considered for solving unconstrained discrete-time optimal control (DTOC) problems, in which the trust-region subproblem can be solved within an acceptable accuracy without explicitly forming the Hessian. The approach is based on the inverse power method for the eigenvalue problem and can handle the hard case. It leads to more efficient algorithms for DTOC problems.

- A. Liao (1997). Some efficient algorithms for unconstrained discrete-time optimal control problems. *Applied Mathematics and Computation*, **87**(2–3), 175–198.

**Summary.** Several algorithms for discrete time optimal control problems are proposed, which combine a modified dogleg algorithm with the differential dynamic programming method or Pantoja's (1988) Newton procedure. These algorithms possess advantages of both the dogleg algorithm and the double-dogleg procedure or the stagewise procedure; i.e., they have strong global and local convergence properties yet remain economical. Numerical results are presented to compare these algorithms and the Coleman and Liao (1995) algorithm.

- C. Lin and J. J. Moré (1999a). Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, **9**(4), 1100–1127.

**Summary.** A trust-region version of Newton's method for bound-constrained problems is proposed that relies on the geometry of the feasible set, not on the particular representation in terms of constraints. The convergence theory holds for linearly constrained problems and yields global and superlinear convergence theory without assuming either strict complementarity or linear independence of the active constraints. The theory also leads to an efficient implementation for large bound-constrained problems.

- C. Lin and J. J. Moré (1999b). Incomplete Cholesky factorizations with limited memory. *SIAM Journal on Scientific Computing*, **21**(1), 24–45.

**Summary.** An incomplete Cholesky factorization is proposed for the solution of large-scale trust-region subproblems and positive definite systems of linear equations. This factorization depends on a parameter  $p$  that specifies the amount of additional memory (in multiples of  $n$ , the dimension of the problem) that is available; there is no need to specify a drop tolerance. The numerical results show that the number of conjugate gradient iterations and the computing time are reduced dramatically for small values of  $p$ . It is also shown that, in contrast with drop tolerance strategies, the new approach is more stable in terms of the number of iterations and memory requirements.

- D. Liu and J. Nocedal (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming, Series B*, **45**(3), 503–528.

- G. Liu, J. Han, and S. Wang (1998). A trust region algorithm for bilevel programming problems. *Chinese Science Bulletin*, **43**(10), 820–824.

**Summary.** A trust-region algorithm is proposed for solving bilevel programming problems where the lower level programming problem is a strongly convex programming problem with linear constraints. This algorithm is based on a trust-region algorithm for nonsmooth unconstrained optimization problems, and its global convergence is established.

- X. Liu and Y. Yuan (1998). A robust trust-region algorithm for solving general nonlinear programming problems. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong.

**Summary.** A trust-region method for general constrained optimization is discussed that uses a composite step technique. The normal step is obtained by solving an ordinary trust-region subproblem, while the tangential step results from a trust-region constrained quadratic program. The method has similarities with those of Burke (1992), Yuan (1995), Dennis, El-Alem, and Maciel (1997), and Dennis and Vicente (1997), but features a new updating rule for the penalty parameter. It is globally convergent, and numerical results indicate that its efficiency is comparable to that of the Harwell Subroutine Library sequential programming code VF02AD.

- F. A. Lootsma (1969). Hessian matrices of penalty functions for solving constrained optimization problems. *Philips Research Reports*, **24**, 322–331.
- P. Lötstedt (1984). Solving the minimal least squares problem subject to bounds on the variables. *BIT*, **24**, 206–224.
- A. Lucia, X. Z. Guo, and X. Wang (1993). Process simulation in the complex-domain. *Aiche Journal*, **39**(3), 461–470.

**Summary.** The asymptotic behaviour of fixed-point methods in the complex domain is studied. Both direct substitution and Newton's method exhibit stable periodic and aperiodic behaviour from real- or complex-valued starting points. Moreover, multiple stable periodic orbits can exist for direct substitution. Traditional trust-region (or dogleg) methods, on the other hand, often terminate at singular points, which correspond to nonzero-valued saddlepoints in the least-squares function that can be arbitrarily far from a solution. Furthermore, the basins of attraction of these singular points are usually dispersed throughout the basin boundaries in the complex domain, clearly illustrating that singular points (via the dogleg strategy) also attract either real- or complex-valued starting points. In this light, an extension of the dogleg strategy to the complex domain, based on a simple norm-reducing, singular point perturbation, is proposed. This extended trust-region method always terminates at a fixed point, even from critical point (worst-case) initial values. Numerical results and geometric illustrations using chemical process simulation examples are presented.

- A. Lucia and J. Xu (1990). Chemical process optimization using Newton-like methods. *Computers and Chemical Engineering*, **14**(2), 119–138.

**Summary.** An algorithm for solving large, sparse quadratic programming (QP) problems that is based on an active-set strategy and a symmetric, indefinite factorization is presented. A simple asymmetric trust-region method is proposed for improving the reliability of successive QP methods. Ill-defined QP subproblems are avoided by adjusting the size of the trust region in an automatic way. It is shown that reliable initial values of the unknown variables and multipliers can be generated automatically using generic problem information, short-cut techniques, and simulation tools. Relevant numerical results and illustrations are presented.

- A. Lucia and J. Xu (1994). Methods of successive quadratic programming. *Computers and Chemical Engineering*, **18**, S211–215.

**Summary.** Simple chemical process examples are constructed that exhibit nondescent in successive quadratic programming as a consequence of the projected indefiniteness of the Hessian matrix of the Lagrangian function. Moreover, in situations where multiple Kuhn–Tucker points for the quadratic programming subproblems exist, the global optimum need not necessarily provide a direction of descent. Thus search for a global solution is unjustified. To circumvent these difficulties, a linear programming-based trust-region method is proposed to guarantee descent for any arbitrary merit function, provided such a direction exists. Geometric illustrations are used to elucidate the main ideas.

- A. Lucia, J. Xu, and K. M. Layn (1996). Nonconvex process optimization. *Computers and Chemical Engineering*, **20**(12), 1375–1398.

**Summary.** Difficulties associated with nonconvexity in successive quadratic programming (SQP) methods are studied. It is shown that projected indefiniteness of the Hessian matrix of the Lagrangian function can (i) place restrictions on the order in which inequalities can be added or deleted from the active set, (ii) generate redundant active sets whose resolution is nontrivial, (iii) give rise to quadratic programming (QP) subproblems that have multiple Kuhn–Tucker points, and (iv) produce nondescent directions in the SQP method that can lead to failure. Related issues concerned with the use of feasible or infeasible starting points for the iterative quadratic programs, forcing positive definiteness to ensure convexity and using iterative methods to solve the linear Kuhn–Tucker conditions associated with the QP subproblems, are studied. An active-set strategy that (i) monitors projected indefiniteness to guide the addition of constraints to the active set, (ii) permits line searching for negative values of the linesearch parameter, and (iii) does not necessarily delete active constraints with incorrect Kuhn–Tucker multipliers is proposed. Constraint redundancy is circumvented using an algorithm that identifies all nontrivial redundant subsets of smallest size and determines which, if any, exchanges are justified. Nondescent in the nonlinear programs is resolved using a linear programming-based trust-region method that guarantees descent regardless of merit function. It is shown that there is no justification for using feasible starting points at the QP level of the calculations, that forcing positive definiteness to ensure convexity can cause termination at undesired solutions, and that the use of iterative methods to solve the linear Kuhn–Tucker equations for the QPs can cause a deterioration in numerical performance. Many small chemical process examples are used to highlight difficulties so that geometric illustrations can be used while heat exchange network design and distillation operations examples are used to show that these same difficulties carry over to the larger problems.

- S. Lucidi (1992). New results on a continuously differentiable exact penalty function. *SIAM Journal on Optimization*, **2**(4), 558–574.
- S. Lucidi, L. Palagi, and M. Roma (1994). Quadratic programs with a quadratic constraint: Characterisation of KKT points and equivalence with an unconstrained problem. Technical Report 24-94, University of Rome “La Sapienza”, Rome, Italy.
- Summary.** This is an earlier version of Lucidi, Palagi, and Roma (1998), containing additional technical detail.
- S. Lucidi, L. Palagi, and M. Roma (1998). On some properties of quadratic programs with a convex quadratic constraint. *SIAM Journal on Optimization*, **8**(1), 105–123.
- Summary.** The problem of minimizing a nonconvex quadratic function with a quadratic constraint is considered, and properties of the problem identified. In particular, (i) given a Karush–Kuhn–Tucker point that is not a global minimizer, it is easy to find a “better” feasible point; and (ii) strict complementarity holds at the local-nonglobal minimum point. It is shown that the original constrained problem is equivalent to the unconstrained minimization of a piecewise quartic merit function. Using this formulation, a second-order necessary condition for global minimum points is given in the nonconvex case. Algorithmic applications are outlined, and preliminary numerical experiments reported.
- D. G. Luenberger (1969). *Optimization by Vector Space Methods*. Wiley, Chichester, England.
- D. G. Luenberger (1984). *Linear and Nonlinear Programming*, second ed. Addison-Wesley, Reading, MA, USA.
- L. Lukšan (1993). Inexact trust region method for large sparse nonlinear least-squares. *Kybernetika*, **29**(4), 305–324.

**Summary.** It is shown that linear least-squares methods based on the least-squares QR algorithm can be used for generation of a trust-region path. This property is a basis for an inexact trust-region method. Numerical experiments suggest that this method is efficient for large sparse nonlinear least squares.

- L. Lukšan (1994). Inexact trust region method for large sparse systems of nonlinear equations. *Journal of Optimization Theory and Applications*, **81**(3), 569–590.

**Summary.** The global convergence of the a trust-region method based on the smoothed conjugate gradients squared algorithm is proved. Numerical experiments indicate that the method is surprisingly convenient for the numerical solution of large sparse systems of nonlinear equations. A modification of the method does not use matrices and can be used for large dense systems of nonlinear equations.

- L. Lukšan (1996a). Combined trust region methods for nonlinear least-squares. *Kybernetika*, **32**(2), 121–138.

**Summary.** Three trust-region variants of the Gauss–Newton method for solving nonlinear least-squares problems are proposed. These comprise a multiple dogleg strategy for dense problems and two combined conjugate gradient Lanczos strategies for sparse problems. Efficiency of these methods is illustrated by extensive numerical experiments.

- L. Lukšan (1996b). Efficient trust region method for nonlinear least-squares. *Kybernetika*, **32**(2), 105–120.

**Summary.** Suitable transformations and decompositions lead to an efficient trust-region method that uses a single factorization at each iteration. This is compared to the optimal locally constrained step that uses more than one decomposition per iteration. Numerical experiments suggest that the former approach is more efficient.

- L. Lukšan (1996c). Hybrid methods for large sparse nonlinear least-squares. *Journal of Optimization Theory and Applications*, **89**(3), 575–595.

- L. Lukšan and J. Vlček (1996). Optimization of dynamical systems. *Kybernetika*, **32**(5), 465–482.

**Summary.** Optimization problems where the objective function is an integral containing the solution of a system of ordinary differential equations are considered. It is shown that optimization methods and methods for initial value problems for ordinary differential equations can be efficiently combined. General procedures for the evaluation of gradients and Hessian matrices are described. An efficient Gauss–Newton-like approximation of the Hessian matrix is derived for the special case when the objective function is an integral of squares. This approximation is used to derive a Gauss–Newton-like trust-region method, for which global and superlinear convergence properties are proved. Finally, several methods are proposed and illustrated by computational experiments.

- L. Lukšan and J. Vlček (1997). Truncated trust region methods based on preconditioned iterative subalgorithms for large sparse systems of nonlinear equations. *Journal of Optimization Theory and Applications*, **95**(3), 637–658.

**Summary.** Globally convergent methods for solving large sparse systems of nonlinear equations with an inexact approximation of the Jacobian matrix are studied. These methods include difference versions of the Newton method and various quasi-Newton methods. A class of trust-region methods is proposed together with a proof of their global convergence, and an implementable globally convergent algorithm is described that can be used as a realization of these methods. Emphasis is put on the application of conjugate gradient-type iterative

methods to the solution of linear subproblems. It is shown that both the GMRES and the smoothed conjugate gradients squared well-preconditioned methods can be used for the construction of globally convergent trust-region methods. The efficiency of the given algorithm is demonstrated computationally by using a large collection of sparse test problems.

- Z. Q. Luo and P. Tseng (1993). Error bounds and convergence analysis of feasible direction methods: A general approach. *Annals of Operations Research*, **46**, 157–178.
- Z. Q. Luo and P. Tseng (1997). A new class of merit functions for the nonlinear complemetarity problem. In M. C. Ferris and J. S. Pang, eds., *Complementarity and Variational Problems: State of the Art*, pp. 204–225, SIAM, Philadelphia, USA.
- S. Lyle and M. Szularz (1994). Local minima of the trust region problem. *Journal of Optimization Theory and Applications*, **80**(1), 117–134.

**Summary.** The minimization of a quadratic form subject to the two-norm constraint is considered. The existence of local minima is investigated. The problem is approached via the dual Lagrangian, and necessary and sufficient conditions for the existence of all local minima are derived. The suitability of the conventional numerical techniques used to solve the problem on processor arrays is examined. Hardware-oriented multisection algorithms are considered, and their efficiency demonstrated on small- to medium-size problems.

- Z. A. Maany (1987). A new algorithm for highly curved constrained optimization. *Mathematical Programming Studies*, **31**, 139–154.
- K. Madsen (1975). An algorithm for the minimax solution of overdetermined systems of nonlinear equations. *Journal of the Institute of Mathematics and Its Applications*, **16**(3), 321–328.

**Summary.** A method for nonlinear min-max in which linear models are minimized subject to an  $\ell_\infty$  trust region.

- R. K. Madayastha and B. Aazhang (1994). An algorithm for training multilayer perceptrons for data classification and function interpolation. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **41**(12), 866–875.

**Summary.** The application of a globally convergent optimization scheme to the training of the multilayer perceptron is discussed. The algorithm combines the conjugate gradient and trust-region algorithms. The potential of the multilayer perceptron, trained using the algorithm, is considered in signal classification in a multiuser communication system, and in approximating the inverse kinematics of a robotic manipulator. It is seen that the multilayer perceptron trained with the trust-region algorithm is able to approximate the desired functions to a greater accuracy than when trained using back propagation. Specifically, in the case of the multiuser communication problem, lower probabilities of error in demodulating a given user's signal, and, in the robotics problem, lower root mean square errors in approximating the inverse kinematics function, are obtained.

- M. K. Mallick (1997). Applications of nonlinear orthogonal distance regression in three-dimensional motion estimation. In S. Van Huffel, ed. *Recent Advances in Total Least-Squares Techniques and Error-in-Variables Modeling*, pp. 273–282, SIAM, Philadelphia, USA.

- O. L. Mangasarian (1979). *Nonlinear Programming*. McGraw-Hill, New York, USA. Reprinted as Classics in Applied Mathematics 10, SIAM, Philadelphia, USA, 1994.
- O. L. Mangasarian (1980). Locally unique solutions of quadratic programs, linear and non-linear complementarity problems. *Mathematical Programming*, **19**(2), 200–212.
- O. L. Mangasarian and S. Fromovitz (1967). The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, **17**, 37–47.
- O. L. Mangasarian and M. V. Solodov (1993). Nonlinear complementarity as unconstrained and constrained minimization. *Mathematical Programming, Series B*, **62**(2), 277–297.
- N. Maratos (1978). Exact penalty function algorithms for finite-dimensional and control optimization problems. Ph.D. thesis, University of London, England.
- D. Marquardt (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, **11**, 431–441.
- Summary.** Taylor-series and steepest-descent methods are sometimes ineffective as algorithms for the least-squares estimation of nonlinear parameters. A maximum neighbourhood method is developed that, in effect, performs an optimum interpolation between Taylor-series and steepest-descent methods. The interpolation is based upon the maximum neighbourhood in which the truncated Taylor series gives an adequate representation of the nonlinear model.
- B. Martinet (1970). Régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'Informatique et de Recherche Opérationnelle*, **4**, 154–159.
- J. M. Martínez (1994). Local minimizers of quadratic functions on Euclidean balls and spheres. *SIAM Journal on Optimization*, **4**(1), 159–176.
- Summary.** A characterization of the local-nonglobal minimizers of a quadratic function defined on a Euclidean ball or sphere is given. It is proven that there exists at most one local-nonglobal minimizer and that the Lagrange multiplier that corresponds to this minimizer is the largest solution of a nonlinear scalar equation. An algorithm is proposed for computing the local-nonglobal minimizer.
- J. M. Martínez (1995). Discrimination by means of a trust region method. *International Journal of Computer Mathematics*, **55**(1–2), 91–103.
- Summary.** The individuals of a population are divided into two groups according to some unknown merit criterion and the problem is considered to determine weights for a set of variables that should be positively correlated with merit in such a way that scores of the individuals in the superior group are above some level, and vice-versa. This situation is modeled as an easy convex optimization problem.
- J. M. Martínez (1997). An algorithm for solving sparse nonlinear least squares problems. *Computing*, **39**(4), 307–325.

**Summary.** A method is given for solving nonlinear least-squares problems, when the Jacobian matrix of the system is large and sparse. The main features of the method are that the Gauss–Newton equation is “partially” solved at each iteration using a preconditioned conjugate gradient algorithm and that the new point is obtained using a two-dimensional trust-region scheme, similar to the one introduced by Bulteau and Vial (1987). Global convergence results and numerical results are presented.

- J. M. Martínez and A. C. Moretti (1997). A trust region method for minimization of nonsmooth functions with linear constraints. *Mathematical Programming*, **76**(3), 431–449.

**Summary.** A trust-region algorithm for minimization of nonsmooth functions with linear constraints is introduced. At each iteration, the objective function is approximated by a model that satisfies assumptions stated by Qi and Sun (1994) for unconstrained nonsmooth optimization. The trust-region iteration begins with the solution of an “easy problem”, as in Martínez and Santos (1995) and Friedlander, Martínez, and Santos (1994a). In practical implementations, the infinity norm is used to define the trust region. Global convergence is established, and numerical experiments for the parameter estimation problem reported.

- J. M. Martínez and S. A. Santos (1995). A trust-region strategy for minimization on arbitrary domains. *Mathematical Programming*, **68**(3), 267–301.

**Summary.** A trust-region method for minimizing a general differentiable function restricted to an arbitrary closed set is presented, and global convergence is proved. The case where the domain is a Euclidean ball is analysed in detail. For this case, numerical experiments that consider a variety of Hessian approximations are presented.

- J. M. Martínez and S. A. Santos (1997). New convergence results on an algorithm for norm constrained regularization and related problems. *RAIRO Recherche Opérationnelle. Operations Research*, **31**(3), 269–294.

**Summary.** The constrained least-squares regularization of nonlinear ill-posed problems is a nonlinear programming problem for which trust-region methods have been developed. It is shown that for one such method, under suitable hypotheses, local (superlinear or quadratic) convergence occurs and every accumulation point is second-order stationary.

- H. Maurer and J. Zowe (1979). First and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems. *Mathematical Programming*, **16**, 98–110.

- D. Mauricio and N. Maculan (1997). A trust region method for zero-one nonlinear programming. *RAIRO Recherche Opérationnelle. Operations Research*, **31**(4), 331–341.

**Summary.** An  $O(n \log n)$  trust-region approximation method to solve 0-1 nonlinear programming is presented. Optimality conditions and numerical results are reported.

- D. Q. Mayne and N. Maratos (1979). A first-order, exact penalty function algorithm for equality constrained optimization problems. *Mathematical Programming*, **16**(3), 303–324.

- D. Q. Mayne and E. Polak (1982). A superlinearly convergent algorithm for constrained optimization problems. *Mathematical Programming Studies*, **16**, 45–61.

- K. B. McAfee, D. M. Gay, R. S. Hozack, R. A. Laudise, G. Schwartz, and W. A. Sunder (1986). Thermodynamic considerations in the synthesis and crystal growth of GaSb. *Journal of Crystal Growth*, **76**(2), 263–271.

**Summary.** The “sticky trust region technique” for Gibbs energy minimization is used to determine the gaseous species and liquid and solid phases present during the synthesis and crystal growth of GaSb. The growth system involves almost thirty species comprising a gaseous phase and nine condensed species.

- K. B. McAfee, D. M. Gay, R. S. Hozack, R. A. Laudise, and W. A. Sunder (1988). Thermodynamic stability and reactivity of AlSb and their relationship to crystal growth. *Journal of Crystal Growth*, **88**(4), 488–498.

**Summary.** The vapor pressure, species concentration, and reactivity of AlSb is modeled thermodynamically using the newly described sticky trust-region technique for free energy minimization. The conditions chosen include oxygen and hydrogen concentrations appropriate to Czochralski crystal growth.

- B. J. McCartin (1998). A model trust-region algorithm utilizing a quadratic interpolant. *Journal of Computational and Applied Mathematics*, **91**(2), 249–259.

**Summary.** A trust-region algorithm for problems in unconstrained optimization and for nonlinear equations utilizing a quadratic interpolant is presented and analysed. This is offered as an alternative to the piecewise-linear interpolant employed in the double-dogleg strategy. A step selection algorithm is presented, along with a summary, with proofs, of its desirable mathematical properties. Numerical results are presented.

- G. P. McCormick (1969). Anti-zig-zagging by bending. *Management Science*, **15**, 315–319.

- G. P. McCormick (1991). The superlinear convergence of a nonlinear primal-dual algorithm. Technical Report OR T-550/91, Department of Operations Research, George Washington University, Washington, DC, USA.

- M. P. McKenna, J. P. Mesirov, and S. A. Zenios (1995). Data parallel quadratic programming on box-constrained problems. *SIAM Journal on Optimization*, **5**(3), 570–589.

- S. Mehrotra and J. Sun (1991). A method of analytic centers for quadratically constrained convex quadratic programs. *SIAM Journal on Numerical Analysis*, **28**, 529–544.

- J. Mentel and H. Anderson (1991). A new kind of parameter-estimation of reactions under dynamic temperature program. *Thermochimica Acta*, **187**(SEP), 121–132.

**Summary.** A trust-region method is applied to a parameter estimation related to kinetic evaluation of thermogravimetry, differential thermal analysis, and differential scanning calorimetry of simple and complex reactions. When all experimental data is included and the differential equation systems used, this proves advantageous compared to the usual linearizing methods. The problems of consecutive and competing reactions as well as those with steady states (enzyme kinetics) are solved satisfactorily if two or more data sets with different heating rates are known. Supplementary use of other analytical methods are recommended.

- J. Mentel, V. Tiller, E. Moller, and D. Haberland (1992). Estimation of parameters in systems of ordinary differential-equations to the determination of kinetic-parameters. *Chemische Technik*, **44**(9), 300–303.

**Summary.** The integral determination of kinetic constants in complex systems is handled as a special case of parameter estimation in systems of differential equations. A trust-region method is used. A stable backward differentiation formula integration routine allows the use of bad initial values. Simulations of different models prove the efficiency of the method even with nonsmooth data.

- R. Mifflin (1975a). Convergence bounds for nonlinear programming algorithms. *Mathematical Programming*, **8**(3), 251–271.

- R. Mifflin (1975b). A superlinearly convergent algorithm for minimization without evaluating derivatives. *Mathematical Programming*, **9**(1), 100–117.

- H. Mine, M. Fukushima, and Y. Tanaka (1984). On the use of epsilon-most-active constraints in an exact penalty function method for nonlinear optimization. *IEEE Transactions on Automatic Control*, **AC-29**(11), 1040–1042.

**Summary.** A globally convergent algorithm for nonlinear programming problems is presented that utilizes the epsilon-most-active constraint strategy in an exact penalty function method with trust region. The algorithm is particularly suitable for problems containing a large number of constraints. Some computational experiments are reported.

- M. Mongeau and A. Sartenaer (1995). Automatic decrease of the penalty parameter in exact penalty function methods. *European Journal of Operational Research*, **83**(3), 686–699.

- R. D. C. Monteiro and I. Adler (1989). Interior path following primal-dual algorithms. 2. Convex quadratic programming. *Mathematical Programming*, **44**(1), 43–66.

- R. D. C. Monteiro and T. Tsuchiya (1998). Global convergence of the affine scaling algorithm for convex quadratic programming. *SIAM Journal on Optimization*, **8**(1), 26–58.

- R. D. C. Monteiro and Y. Wang (1998). Trust region affine scaling algorithms for linearly constrained convex and concave programs. *Mathematical Programming*, **80**(3), 283–310.

**Summary.** A trust-region affine-scaling algorithm for solving the linearly constrained convex or concave programming problem is presented. Under primal nondegeneracy assumption, every accumulation point of the sequence generated by the algorithm satisfies the first-order necessary condition. For a special class of convex or concave functions satisfying a certain invariance condition on their Hessians, the sequence of iterates and objective function values converge R-linearly and Q-linearly, respectively. Moreover, under primal nondegeneracy and for this class of objective functions, the limit point of the sequence of iterates satisfies the first- and second-order necessary conditions.

- J. J. Moré (1978). The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, ed., *Numerical Analysis*, Dundee 1977, pp. 105–116, Lecture Notes in Mathematics 630, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** A robust implementation of the Levenberg–Morrison–Marquardt algorithm for nonlinear least squares is discussed. The proposed method is shown to have strong convergence properties. In addition to robustness, the main features are the proper use of implicitly scaled variables and the choice of the Levenberg–Morrison–Marquardt parameter via a scheme due to Hebden (1973). Numerical results illustrating the behaviour of this implementation are presented.

- J. J. Moré (1983). Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, eds., *Mathematical Programming: The State of the Art*, pp. 258–287, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** The theoretical and practical results available for trust-region methods in systems of nonlinear equations, nonlinear estimation problems, and large-scale optimization are surveyed and their relevance to the implementation of trust-region methods discussed.

- J. J. Moré (1988). Trust regions and projected gradients. In M. Iri and K. Yajima, eds., *System Modelling and Optimization*, pp. 1–13, Lecture Notes in Control and Information Sciences 113, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** It is shown how the ideas from the gradient-projection method combine with trust-region methods, and an indication of the powerful convergence results that are available for gradient-projection algorithms is given.

- J. J. Moré (1993). Generalizations of the trust region problem. *Optimization Methods and Software*, **2**(3), 189–209.

**Summary.** The trust-region subproblem is generalized by allowing a general quadratic constraint. The main results are a characterization of the global minimizer of the generalized trust-region problem, and the development of an algorithm that finds an approximate global minimizer in a finite number of iterations.

- J. J. Moré, B. S. Garbow, and K. E. Hillstrom (1980). User guide for MINPACK-1. Technical Report 80–74, Argonne National Laboratory, Argonne, IL, USA.

- J. J. Moré, B. S. Garbow, and K. E. Hillstrom (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, **7**(1), 17–41.

- J. J. Moré and D. C. Sorensen (1983). Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, **4**(3), 553–572.

**Summary.** An algorithm for minimizing a quadratic function subject to an ellipsoidal constraint is proposed. This algorithm is guaranteed to produce a nearly optimal solution in a finite number of iterations. The use of this algorithm in a trust-region Newton's method is also considered. In particular, it is shown that under reasonable assumptions the sequence generated by Newton's method has a limit point that satisfies first- and second-order necessary conditions for a minimizer. Numerical results for GQTPAR, a Fortran implementation of the algorithm, show that it is quite successful in a trust-region method. In these tests, a call to GQTPAR only required 1.6 iterations on average.

- J. J. Moré and D. C. Sorensen (1984). Newton's method. In G. H. Golub, ed., *Studies in Numerical Analysis*, pp. 29–82, MAA Studies in Mathematics 24, American Mathematical Society, Providence, RI, USA.

**Summary.** The linesearch and trust-region approaches for unconstrained optimization are discussed, and some of the recent developments related to Newton's method are presented. In particular, several variations on Newton's method that are appropriate for large-scale problems are explored, and it is shown how quasi-Newton methods can be derived quite naturally from Newton's method.

J. J. Moré and G. Toraldo (1991). On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, **1**(1), 93–113.

J. J. Moré and S. J. Wright (1993). *Optimization Software Guide*. Frontiers in Applied Mathematics 14, SIAM, Philadelphia, USA.

J. J. Moreau (1962). Décomposition orthogonale d'un espace Hilbertien selon deux cônes mutuellement polaires. *Comptes-Rendus de l'Académie des Sciences (Paris)*, **255**, 238–240.

D. D. Morrison (1960). Methods for nonlinear least squares problems and convergence proofs. In J. Lorell and F. Yagi, eds., *Proceedings of the Seminar on Tracking Programs and Orbit Determination*, pp. 1–9, Jet Propulsion Laboratory, Pasadena, CA, USA.

**Summary.** Least-squares estimation of missile trajectory is considered in the context of lunar and interplanetary flights as well as earthbound satellite tracking. A least-squares subroutine is described in which convergence can be assumed by limiting the amount any variable can change in one iteration. A method is given that allows the computation of a quadratic model of the objective function within a sphere using a single linear system depending on a parameter. Monotonicity of the optimal model value as a function of this parameter is proved.

H. Mukai and E. Polak (1975). A quadratically convergent primal-dual algorithm with global convergence properties for solving optimization problems with inequality constraints. *Mathematical Programming*, **9**(3), 336–349.

K. Mukai, K. Tatsumi, and M. Fukushima (1998). An approximation algorithm for quadratic cost 0-1 mixed integer programming problems. *Transactions of the Institute of Electronics, Information and Communication Engineers A*, **J81-A**(4), 649–657.

**Summary.** The quadratic cost 0-1 mixed integer programming problem is formulated as a two-level programming problem that consists of the lower level continuous quadratic programming problem with 0-1 variables being fixed and the upper level nonlinear 0-1 programming problem. An approximation algorithm for solving the upper level 0-1 programming problem is proposed that approximately solves a subproblem obtained by linearizing the objective function at a current point. To guarantee the descent property of the generated sequence, a trust-region technique adaptively controls a penalty constant in the objective function of the subproblem. To solve subproblems, a Hopfield network is applied with a new transition rule that allows a temporary state transition based on the variable depth method. Some numerical experiments for a location-transportation problem with quadratic costs indicate that the proposed algorithm is practically effective.

W. Murray (1969). An algorithm for constrained minimization. In R. Fletcher, ed., *Optimization*, pp. 189–196, Academic Press, London.

- W. Murray (1971a). An algorithm for finding a local minimum of an indefinite quadratic program. Technical Report NAC 1, National Physical Laboratory, London, England.
- W. Murray (1971b). Analytical expressions for eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. *Journal of Optimization Theory and Applications*, **7**, 189–196.
- W. Murray (1992). Ill-conditioning in barrier methods. In *Advances in Numerical Partial Differential Equations and Optimization, Proceedings of the Sixth Mexico-United States Workshop*, SIAM, Philadelphia, USA.
- W. Murray and F. J. Prieto (1995). A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM Journal on Optimization*, **5**(3), 590–640.
- W. Murray and M. H. Wright (1978). Project Lagrangian methods based on the trajectories of penalty and barrier functions. Technical Report SOL78-23, Department of Operations Research, Stanford University, Stanford, CA, USA.
- W. Murray and M. H. Wright (1982). Computation of the search direction in constrained optimization algorithms. *Mathematical Programming Studies*, **16**, 62–83.
- B. A. Murtagh (1981). *Advanced Linear Programming*. McGraw-Hill, New York, USA.
- K. G. Murty and S. N. Kabadi (1987). Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, **39**(2), 117–129.
- N. Nabona (1987). Computational results of Newton's method with search along an approximate hook step curve for unconstrained minimization. In J. P. Vilaplana and L. F. Escudero, eds., *Actas I Seminario Internacional de Investigacion Operativa del País Vasco*, pp. 21–54, Argitaparen Zerbitzua Euskal Herriko Unibertsitatea, Bilbao, Spain.
- Summary.** Numerical experience is presented for an unconstrained optimization method in which a trust-region step is computed by minimizing the model along a Bezier curve that approximates the trajectory of exact minimizers as a function of the trust-region radius.
- NAG (1998). *Fortran Library Mark 18*. NAG, Oxford, England.
- A. Nagurney (1993). *Network Economics: A Variational Inequality Approach*. Advances in Computational Economics, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- S. G. Nash (1984). Newton-type minimization via the Lanczos method. *SIAM Journal on Numerical Analysis*, **21**(4), 770–788.
- S. G. Nash (1985). Preconditioning of truncated Newton methods. *SIAM Journal on Scientific and Statistical Computing*, **6**(3), 599–618.

- S. G. Nash and J. Nocedal (1991). A numerical study of the limited memory BFGS method and the truncated-Newton method for large-scale optimization. *SIAM Journal on Optimization*, **1**(3), 358–372.
- S. G. Nash and A. Sofer (1990). Assessing a search direction within a truncated-Newton method. *Operations Research Letters*, **9**(4), 219–221.
- S. G. Nash and A. Sofer (1993). A barrier method for large-scale constrained optimization. *ORSA Journal on Computing*, **5**(1), 40–53.
- S. G. Nash and A. Sofer (1998). Why extrapolation helps in barrier methods. Technical Report, Operations Research and Engineering Department, George Mason University, Fairfax, VA, USA.
- J. A. Nelder and R. Mead (1965). A simplex method for function minimization. *Computer Journal*, **7**, 308–313.
- S. A. Nelson and P. Y. Papalambros (1998). A modified trust region algorithm for hierarchical nlp. *Structural Optimization*, **16**(1), 19–28.
- Summary.** Modifications are made to a trust-region algorithm to take advantage of the hierarchical structure in large-scale optimization problems without compromising the theoretical properties of the original algorithm.
- S. A. Nelson and P. Y. Papalambros (1999). The use of trust region algorithms to exploit discrepancies in function computation time within optimization models. *Journal of Mechanical Design*, **121**(4), 552–556.
- Summary.** Yuan's (1995) trust-region algorithm is modified so that it is better able to use function evaluations whose computational cost may vary enormously. This technique is applied to valve event optimization of an internal combustion engine.
- A. Nemirovskii and K. Scheinberg (1996). Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic problems. *Mathematical Programming, Series A*, **72**(3), 273–289.
- Y. Nesterov (1998). Semidefinite relaxation and nonconvex quadratic optimization. *Optimization Software and Methods*, **9**(1–3), 141–160.
- Y. Nesterov and A. Nemirovskii (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, USA.
- Y. Nesterov and M. J. Todd (1998). Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, **8**(2), 324–364.
- B. Noble and J. W. Daniel (1977). *Applied Linear Algebra*, second ed. Prentice-Hall, Englewood Cliffs, NJ, USA.
- J. Nocedal (1980). Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, **35**, 773–782.

- J. Nocedal (1984). Trust region algorithms for solving large systems of nonlinear equations. In W. Liu, T. Belytschko, and K. C. Park, eds., *Innovative Methods for Nonlinear Problems*, pp. 93–102, Pineridge Press, Swansea, Wales.

**Summary.** Algorithms for solving large sparse systems of nonlinear equations are reviewed, with emphasis on trust-region methods. An algorithm is described in which the  $\ell_1$  norm of the linearized equations is minimized within an  $\ell_\infty$ -norm trust region. The resulting subproblems are solved by means of sparse linear programming techniques.

- J. Nocedal (1986). Viewing the conjugate-gradient algorithm as a trust region method. In J. P. Hennart, ed., *Numerical Analysis*, pp. 118–126, Lecture Notes in Mathematics 1230, Springer-Verlag, Heidelberg, Berlin, New York.

**Summary.** It is shown how a trust-region subproblem, based upon a memoryless secant-updating formula, may be solved very efficiently. A connection is made between such subproblems and the conjugate gradient method.

- J. Nocedal and M. L. Overton (1985). Projected Hessian updating algorithms for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, **22**, 821–850.

- J. Nocedal and Y. Yuan (1998). Combining trust region and line search techniques. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 153–176, Kluwer Academic Publishers, Dordrecht, the Netherlands.

**Summary.** An algorithm for nonlinear optimization that employs both trust-region techniques and linesearches is proposed. This algorithm does not resolve the subproblem if the trial step results in an increase in the objective function, but instead performs a backtracking linesearch from the failed point. Backtracking can be done along a straight line or along a curved path. It is shown that the algorithm preserves the strong convergence properties of trust-region methods. Numerical results are presented.

- Y. Notay (1993). On the convergence rate of the conjugate gradients in presence of rounding errors. *Numerische Mathematik*, **65**(3), 301–317.

- D. P. O'Leary (1980). A generalized conjugate gradient algorithm for solving a class of quadratic programming problems. *Linear Algebra and Its Applications*, **34**, 371–399.

- E. O. Omokoko (1989). Trust region algorithms for optimization with nonlinear equality and inequality constraints. Ph.D. thesis, University of Colorado, Boulder, CO, USA.

**Summary.** Trust-region algorithms for the general nonlinear optimization problem are presented. These handle both the equality- and inequality-constrained cases. The algorithms use the sequential quadratic programming approach combined with the trust-region technique. A model subproblem is defined in which a quadratic approximation of the Lagrangian is minimized subject to modified relaxed linearizations of the problem nonlinear constraints and a trust-region constraint. Inequality constraints are handled by a compromise between an active set strategy and inequality-constrained quadratic programming subproblem solution techniques. A local convergence analysis is presented. Numerical tests indicate that the proposed methods are very robust and reasonably efficient.

- J. M. Ortega and W. C. Rheinboldt (1970). *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, London.

- M. R. Osborne (1976). Nonlinear least squares—the Levenberg–Marquardt algorithm revisited. *Journal of the Australian Mathematical Society, Series B*, **19**, 343–357.
- Summary.** An easily implemented rule for choosing the Levenberg–Morrison–Marquardt parameter is described. A satisfactory convergence result is established.
- M. R. Osborne (1985). *Finite Algorithms for Optimization and Data Analysis*. Wiley, Chichester, England.
- M. R. Osborne (1987). Estimating nonlinear models by maximum likelihood for the exponential family. *SIAM Journal on Scientific and Statistical Computing*, **8**(3), 446–456.
- Summary.** Trust-region methods for solving nonlinear least-squares problems are adapted to maximize likelihoods based on the exponential family, while preserving their theoretical properties.
- M. R. Osborne (1998). Variable selection and control in least squares problems. Technical Report MRR 047-98, Centre for Mathematics and Its Applications, School of Mathematical Sciences, Australian National University, Canberra, Australia.
- Summary.** Results are reviewed showing that the  $\ell_1$  trust-region method provides a form of variable selection for least-squares problems, and computational methods are discussed.
- J. Outrata, J. Zowe, and H. Schramm (1991). Bundle trust methods: Fortran codes for nondifferentiable optimization. Technical Report 269, Deutsche Forschungsgemeinschaft, Germany.
- C. C. Paige (1971). The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, University of London, England.
- C. C. Paige and M. A. Saunders (1975). Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, **12**(4), 617–629.
- C. C. Paige and M. A. Saunders (1982). LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, **8**, 43–71.
- J. S. Pang (1981). An equivalence between two algorithms for quadratic programming. *Mathematical Programming*, **20**(2), 152–165.
- J. S. Pang (1995). Complementarity problems. In R. Horst and P. Pardalos, eds., *Handbook of Global Optimization*, pp. 271–338, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- E. Panier and A. L. Tits (1991). Avoiding the Maratos effect by means of a nonmonotone linesearch I. General constrained problems. *SIAM Journal on Numerical Analysis*, **28**, 1183–1195.
- E. Panier, A. L. Tits, and J. N. Herskovits (1988). A QP-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization*, **26**(4), 788–811.

- J. F. A. Pantoja (1988). Differential dynamic-programming and Newton method. *International Journal of Control*, **47**(5), 1539–1553.
- J. F. A. Pantoja and D. Q. Mayne (1991). Exact penalty functions with simple updating of the penalty parameter. *Journal of Optimization Theory and Applications*, **69**, 441–467.
- C. H. Papadimitriou and K. Steiglitz (1982). *Combinatorial Optimization*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- B. N. Parlett (1980). *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, NJ, USA. Reprinted as Classics in Applied Mathematics 20, SIAM, Philadelphia, USA, 1998.
- B. N. Parlett and J. K. Reid (1981). Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems. *Journal of the Institute of Mathematics and Its Applications*, **1**, 135–155.
- M. Patriksson (1994). *The Traffic Assignment Problem: Models and Methods*. VSP, Utrecht, the Netherlands.
- M. Patriksson (1998). *Nonlinear Programming and Variational Inequality Problems, a Unified Approach*. Kluwer Academic Publishers, Dordrecht, the Netherlands.
- J. Peng (1996). Unconstrained methods for generalized nonlinear complementarity and variational inequality problems. *Journal of Computational Mathematics*, **14**(2), 99–107.
- Summary.** Unconstrained methods for the generalized nonlinear complementarity problem and variational inequalities are constructed. Properties of the corresponding unconstrained optimization problem are studied. These methods are applied to the subproblems in trust-region methods, and their interrelationships are studied. Numerical results are presented.
- J. Peng (1998). A smoothing function and its applications. In M. Fukushima and L. Qi, eds., *Reformulation: Nonsmooth, Piecewise Smooth, Semismooth and Smoothing Methods*, pp. 293–316, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- J. Peng and Y. Yuan (1997). Optimality conditions for the minimization of a quadratic with two quadratic constraints. *SIAM Journal on Optimization*, **7**(3), 579–594.
- Summary.** The trust-region subproblem is generalized by allowing two general quadratic constraints. Conditions and properties of its solution are discussed.
- A. Perry (1976). A modified conjugate gradient algorithm. Technical Report 229, Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, IL, USA.
- L. R. Petzold, Y. H. Ren, and T. Maly (1997). Regularization of higher-index differential-algebraic equations with rank-deficient constraints. *SIAM Journal on Scientific Computing*, **18**(3), 753–774.

**Summary.** Several regularizations for higher index differential-algebraic equations with rank-deficient or singular constraints are presented. These types of problems arise in the solution of constrained mechanical systems, when a mechanism's trajectory passes through or near a kinematic singularity. Regularizations for these problems are derived that are based on minimization of the norm of the constraints. They are analogous to trust-region methods. Convergence results are given and numerical experiments are presented.

- T. Pham Dinh and H. A. Le Thi (1995). Lagrangian stability and global optimality on nonconvex quadratic minimization over Euclidean balls and spheres. *Journal of Convex Analysis*, **2**(1–2), 263–276.

**Summary.** Stability of the Lagrangian duality in nonconvex quadratic minimization over Euclidean balls and spheres is proved. Global optimality conditions for these problems are deduced together with the detailed descriptions of the structure of their solution sets.

- T. Pham Dinh and H. A. Le Thi (1998). D.C. optimization algorithm for solving the trust-region subproblem. *SIAM Journal on Optimization*, **8**(2), 476–505.

**Summary.** A framework for DC (difference of convex functions) optimization is presented, including DC duality, local and global optimality, the DC algorithm (DCA), and its application to solving the trust-region problem, which only requires matrix-vector products. In practice, the DCA converges to a global solution, which may be checked using the implicitly restarted Lanczos method of Sorensen. If a nonglobal solution is found, a procedure is proposed that finds a feasible point having a smaller objective value at which the DCA may then be restarted. It is proved that, in the nonconvex case, the DCA needs at most  $2m + 2$  restarts to converge to a global solution, where  $m$  is the number of distinct negative eigenvalues of the Hessian. The robustness and efficiency of the DCA is illustrated by numerical experiments.

- T. Pham Dinh, T. Q. Phong, R. Horaud, and L. Quan (1997). Stability of Lagrangian duality for nonconvex quadratic programming. solution methods and applications in computer vision. *RAIRO Modélisation Mathématique et Analyse Numérique*, **31**(1), 57–90.

**Summary.** The problem of minimizing a quadratic form over a ball is considered. The stability of Lagrangian duality is established and complete characterizations of a global optimal solution are given. Two solution methods are deduced with application to the trust-region subproblem. Mathematical models of some important problems encountered in computer vision are discussed, which can be formulated as a minimization of a sum of squares of nonlinear functions. A practical trust-region-based algorithm is proposed for the nonlinear least-squares problem which seems to be well suited to computer vision applications.

- T. Pham Dinh and S. Wang (1990). Training multi-layered neural network with a trust-region based algorithm. *RAIRO Modélisation Mathématique et Analyse Numérique*, **24**(4), 523–553.

**Summary.** The problem of training a neural network is modelled as an optimization problem, and gradient back propagation, the most commonly used training method, is described. A trust-region algorithm is proposed. Experimental results show that the trust-region algorithm is much faster and more robust than gradient back propagation.

- E. Phan huy Hao (1982). Quadratically constrained quadratic programming: Some applications and a method for solution. *Zeitschrift für Operations Research*, **26**(3), 105–119.

- T. Q. Phong, R. Horaud, A. Yassine, and T. Pham Dinh (1995). Object pose from 2-D to 3-D point and line correspondences. *International Journal of Computer Vision*, **15**(3), 225–243.

**Summary.** See also, by the same authors, “Optimal Estimation of Object Pose from a Single Perspective View”, in *Proceedings Fourth International Conference on Computer Vision*, pp. 534–539, IEEE Computer Society Press, Los Alamitos, CA, USA, 1993. A method for robustly and accurately estimating the rotation and translation between a camera and a three-dimensional object from point and line correspondences is given. An error function is defined and then minimized. This function is quadratic if rotation and translation are represented with a dual number quaternion. Details are given of a trust-region optimization method that compares favourably with Newton’s method, which has been used extensively to solve the problem, with the Faugeras–Toscani linear method for calibrating a camera, and with the Levenberg–Morrison–Marquardt nonlinear optimization method. Experimental results are presented that demonstrate the robustness of the method with respect to image noise and matching errors.

- J. A. Piepmeyer, G. V. McMurray, and H. Lipkin (1998). Tracking a moving target with model independent visual servoing: A predictive estimation approach. In *Proceedings 1998 IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2652–2657, IEEE, New York, USA.

**Summary.** Target tracking by model independent visual servo control is achieved by augmenting quasi-Newton trust-region control with target prediction. Model independent visual servo control is defined using visual feedback to control the robot without precise kinematic and camera models. The use of predictive filters to improve the performance of the control algorithm for linear and circular target motions is demonstrated. The results show a performance of the same order of magnitude as compared to some model-based visual servo control research. Certain limitations to the algorithm are discussed.

- T. Pietrzykowski (1969). An exact potential method for constrained maxima. *SIAM Journal on Numerical Analysis*, **6**(2), 299–304.

- T. D. Plantenga (1999). A trust-region method for nonlinear programming based on primal interior point techniques. *SIAM Journal on Scientific Computing*, **20**(1), 282–305.

**Summary.** A trust-region method for large-scale optimization problems with nonlinear equality and inequality constraints is described. The algorithm employs interior-point techniques from linear programming, adapting them for more general nonlinear problems. An implementation, based entirely on sparse matrix methods, is described. The software handles infeasible start points, identifies the active set of constraints at a solution, and can use second derivative information. Numerical results are reported for large and small problems and a comparison made with other large-scale codes.

- E. Polak (1997). *Optimization. Algorithms and Consistent Approximations*. Applied Mathematical Sciences 124, Springer-Verlag, Heidelberg, Berlin, New York.

- E. Polak and A. L. Tits (1980). A globally convergent implementable multiplier method with automatic penalty limitation. *Applied Mathematics and Optimization*, **6**, 335–360.

- R. Poliquin and L. Qi (1995). Iteration functions in some nonsmooth optimization algorithms. *Mathematics of Operations Research*, **20**(2), 479–496.

**Summary.** Properties of iteration functions, such as their existence and calculus, arising in model trust-region algorithms for nonsmooth problems are analysed. It is shown that a locally Lipschitzian function has a Pang–Han–Rangaraj iteration function only when the function is

- pseudoregular (in the sense of Borwein) and that a subsmooth (lower- $C^1$ ) function always has a Pang–Han–Rangaraj iteration function.
- S. Poljack, F. Rendl, and H. Wolkowicz (1995). A recipe for semidefinite relaxation for  $(0, 1)$ -quadratic programming. *Journal of Global Optimization*, **7**(1), 51–73.
- Summary.** Various relaxations of  $(0, 1)$ -quadratic programming problems are reviewed, including semidefinite programs, parametric trust-region problems, and concave quadratic maximization. All lead to efficiently solvable problems. Using Lagrangian duality, equivalence of the relaxations is proved in a unified way. The approach is extended to the case where equality constraints are present. It is shown how this technique can be applied to the quadratic assignment problem, the graph partition problem, and the max-clique problem. The relaxation is the best possible among all quadratic majorants with zero trace.
- S. Poljack and H. Wolkowicz (1995). Convex relaxations of  $(0, 1)$ -quadratic programming. *Mathematics of Operations Research*, **20**(3), 550–561.
- B. T. Polyak (1969). The conjugate gradient method in extremal problems. *USSR Computational Mathematics and Mathematical Physics*, **9**, 94–112.
- R. Polyak (1982). Smooth optimization methods for solving nonlinear extremal and equilibrium problems with constraints. Presentation at the IXth International Symposium on Mathematical Programming, Bonn, Germany.
- R. Polyak (1992). Modified barrier functions (theory and methods). *Mathematical Programming*, **54**(2), 177–222.
- D. B. Ponceleón (1990). Barrier methods for large-scale quadratic programming. Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, USA.
- F. Pornbacher, U. Fichter, G. Muller-Liebler, and H. Zapf (1990). A new method for an efficient optimization of MOS transistor models. In 1990 *IEEE International Symposium on Circuits and Systems*, Vol. 1, pp. 81–84, IEEE, New York, USA.
- Summary.** Two methods are described for accurate optimization of complex transistor models. The first focuses on sample reduction before the optimization process. An algorithm is described that allows a reduction of the number of samples by a factor of 10 to 20. The second method is a trust-region optimization algorithm, especially designed for this application. Industrial examples demonstrate the quality of the algorithms.
- M. J. D. Powell (1969). A method for nonlinear constraints in minimization problems. In R. Fletcher, ed., *Optimization*, pp. 283–298, Academic Press, London.
- M. J. D. Powell (1970a). A Fortran subroutine for unconstrained minimization requiring first derivatives of the objective function. Technical Report R-6469, AERE Harwell Laboratory, Harwell, Oxfordshire, England.
- Summary.** Details of the implementation of the algorithm described in Powell (1970d) are provided and numerical experiments are discussed. The Fortran code is given in an appendix.
- M. J. D. Powell (1970b). A Fortran subroutine for solving systems of nonlinear algebraic equations. In P. Rabinowitz, ed., *Numerical Methods for Nonlinear Algebraic Equations*, pp. 115–161, Gordon and Breach, London.

**Summary.** Details of the implementation of the algorithm described in Powell (1970c) are provided and numerical experiments are discussed. The Fortran code is given in an appendix.

- M. J. D. Powell (1970c). A hybrid method for nonlinear equations. In P. Rabinowitz, ed., *Numerical Methods for Nonlinear Algebraic Equations*, pp. 87–114, Gordon and Breach, London.

**Summary.** An algorithm for solving systems of nonlinear equations, that does not require the evaluation of the Jacobian of the system, is described. Instead, the derivatives are approximated by Broyden's quasi-Newton formula. The algorithm uses a Levenberg–Morrison–Marquardt procedure for computing a new iterate, together with a safeguarding technique that prevents any tendencies towards linear independence of steps. Convergence of the algorithm is proved to either a solution of the system or a local minimum of the norm of its residual.

- M. J. D. Powell (1970d). A new algorithm for unconstrained optimization. In J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., *Nonlinear Programming*, pp. 31–65, Academic Press, London.

**Summary.** A trust-region algorithm is described for unconstrained smooth minimization. Convergence theorems are given that impose very mild conditions on the objective function. These theorems, together with some numerical results, indicate that the method may be preferable to then-current algorithms for solving unconstrained minimization problems.

- M. J. D. Powell (1975). Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., *Nonlinear Programming*, 2, pp. 1–27, Academic Press, London.

- M. J. D. Powell (1978). A fast algorithm for nonlinearly constrained optimization calculations. In G. A. Watson, ed., *Numerical Analysis, Dundee 1977*, pp. 144–157, Lecture Notes in Mathematics 630, Springer-Verlag, Heidelberg, Berlin, New York.

- M. J. D. Powell (1981a). *Approximation Theory and Methods*. Cambridge University Press, Cambridge, England.

- M. J. D. Powell (1981b). An upper triangular matrix method for quadratic programming. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., *Nonlinear Programming*, 2, Academic Press, London.

- M. J. D. Powell (1983). General algorithms for discrete nonlinear approximation calculations. Technical Report DAMTP/NA2, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England.

- M. J. D. Powell (1984). On the global convergence of trust region algorithms for unconstrained minimization. *Mathematical Programming*, **29**(3), 297–303.

**Summary.** Global convergence for trust-region methods in unconstrained optimization is obtained in the case when there is a bound on the Hessian approximations that depends linearly on the iteration number.

- M. J. D. Powell (1985). On the quadratic-programming algorithm of Goldfarb and Idnani. *Mathematical Programming Studies*, **25**, 46–61.

- M. J. D. Powell (1987). Methods for nonlinear constraints in optimization calculations. In A. Iserles and M. J. D. Powell, eds., *The State of the Art in Numerical Analysis*, pp. 325–358, Oxford University Press, Oxford, England.
- M. J. D. Powell (1993). Log barrier methods for semi-infinite programming calculations. In E. A. Lipitakis, ed., *Advances on Computer Mathematics and Its Applications*, pp. 1–21, World Scientific Publishers, Singapore.
- M. J. D. Powell (1994a). A direct search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J. P. Hennart, eds., *Advances in Optimization and Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, Vol. 275, pp. 51–67, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- M. J. D. Powell (1994b). A direct search optimization method that models the objective by quadratic interpolation. Presentation at the 5th Stockholm Optimization Days, Stockholm, Sweden.
- M. J. D. Powell (1996). Trust region methods that employ quadratic interpolation to the objective function. Presentation at the Fifth SIAM Conference on Optimization, Victoria, BC, Canada.
- M. J. D. Powell (1998a). Direct search algorithms for optimization calculations. *Acta Numerica*, **7**, 287–336.
- M. J. D. Powell (1998b). A quadratic model trust region method for unconstrained minimization without derivatives. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong.
- Summary.** A derivative-free trust-region method for unconstrained optimization is described that uses quadratic interpolation models. These are chosen to be a linear combination of the Lagrange fundamental polynomials associated with the interpolation problem. It is shown that the coefficients of these polynomials can be updated from iteration to iteration in a numerically stable way. The method also uses a separate radius to control the distance between interpolation points.
- M. J. D. Powell (1998c). The use of band matrices for second derivative approximations in trust region algorithms. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 3–28, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Summary.** In many trust-region algorithms, each iteration seeks a vector  $d \in \mathbb{R}^n$  that solves the linear system of equations  $(B + \lambda I)d = -g$ , where  $B$  is a symmetric estimate of a second derivative matrix,  $g$  is a known gradient vector, and  $\lambda$  is a parameter that controls the length of  $d$ . Several values of  $\lambda$  may be tried on each iteration, and, when there is no helpful sparsity in  $B$ , it is usual for each solution to require  $O(n^3)$  operations. However, if an orthogonal matrix  $\Omega$  is available such that  $M = \Omega^T B \Omega$  is an  $n \times n$  matrix of bandwidth  $2s+1$ , then  $\Omega^T d$  can be calculated in only  $O(ns^2)$  operations for each new  $\lambda$  by writing the system in the form  $(M + \lambda I)(\Omega^T d) = -\Omega^T g$ . Unfortunately, it is shown that the construction of  $M$  and  $\Omega$  from  $B$  is usually more expensive than the solution of the original system, but in variable metric and quasi-Newton algorithms for unconstrained optimization, each iteration changes  $B$  by a matrix whose rank is at most 2, and then updating techniques can be applied to  $\Omega$ . Thus it is possible to reduce the average work per iteration from  $O(n^3)$  to  $O(n^{7/3})$  operations. Here

the elements of each orthogonal matrix are calculated explicitly, but instead one can express the orthogonal matrix updates as products of Givens rotations, which allows the average work per iteration to be only  $O(n^{11/5})$  operations. Details of procedures that achieve these savings are described, and the  $O(n^{7/3})$  complexity is confirmed by numerical results.

M. J. D. Powell, editor (1982). *Nonlinear Optimization 1981*, Academic Press, London.

M. J. D. Powell and Ph. L. Toint (1979). On the estimation of sparse Hessian matrices. *SIAM Journal on Numerical Analysis*, **16**(6), 1060–1074.

M. J. D. Powell and Y. Yuan (1986). A recursive quadratic-programming algorithm that uses differentiable exact penalty-functions. *Mathematical Programming*, **35**(3), 265–278.

M. J. D. Powell and Y. Yuan (1990). A trust region algorithm for equality constrained optimization. *Mathematical Programming*, **49**(2), 189–213.

**Summary.** A trust-region algorithm for equality constrained optimization is proposed that employs a differentiable exact penalty function. Under certain conditions, global and local superlinear convergence results are established.

A. I. Propoi and A. V. Pukhlikov (1993). Newton stochastic method in nonlinear extremal problems. *Automation and Remote Control*, **54**(4, Part 1), 605–613.

**Summary.** A random search procedure is explored in the context of trust-region methods.

B. N. Pschenichny (1970). Algorithms for general problems of mathematical programming. *Kibernetika*, **6**, 120–125.

M. Psiaki and K. Park (1995). Augmented Lagrangian nonlinear-programming algorithm that uses SQP and trust region techniques. *Journal of Optimization Theory and Applications*, **86**(2), 311–325.

**Summary.** An augmented Lagrangian nonlinear programming algorithm is developed. The algorithm consists of three nested loops. The outer loop estimates the Karush–Kuhn–Tucker multipliers at a rapid linear rate of convergence. The middle loop minimizes the augmented Lagrangian function for fixed multipliers. This loop uses the sequential quadratic programming technique with a box trust-region stepsize restriction. The inner loop solves a single quadratic program. Slack variables and a constrained form of the fixed-multiplier middle-loop problem work together with curved linesearches in the inner-loop problem to allow large penalty weights for rapid outer-loop convergence. The inner-loop quadratic programs include quadratic constraint terms, which complicate the inner loop, but speed the middle-loop progress when the constraint curvature is large.

L. Qi (1995). Trust region algorithms for solving nonsmooth equations. *SIAM Journal on Optimization*, **5**(1), 219–230.

**Summary.** Two globally convergent trust-region algorithms are presented for solving non-smooth equations in the case where the functions are only locally Lipschitzian. The first algorithm is an extension of the classic Levenberg–Morrison–Marquardt method, obtained by approximating the locally Lipschitzian function with a smooth (quadratic) function and using the derivative of the smooth function in the algorithm wherever a derivative is needed. Global convergence is established under a regularity condition. In the second algorithm, successive smooth quadratic approximation functions and their derivatives are used. Global convergence for the second algorithm is established under mild assumptions.

- L. Qi and J. Sun (1994). A trust region algorithm for minimization of locally Lipschitzian functions. *Mathematical Programming*, **66**(1), 25–43.

**Summary.** The classical trust-region algorithm for smooth nonlinear programs is extended to the nonsmooth case where the objective function is only locally Lipschitzian. At each iteration, an objective function that uses both first- and second-order information is minimized over a trust region. The term that carries the first-order information is an iteration function that may not explicitly depend on subgradients or directional derivatives. It is proved that the algorithm is globally convergent. Applications of the algorithm to various nonsmooth optimization problems are discussed.

- J. K. Reid (1971). On the method of conjugate gradients for the solution of large sparse linear equations. In J. K. Reid, ed., *Large Sparse Sets of Linear Equations*, pp. 231–254, Academic Press, London.

- J. K. Reid (1973). Least squares solution of sparse systems of non-linear equations by a modified Marquardt algorithm. In D. M. Himmelblau, ed., *Decomposition of Large-Scale Problems*, pp. 437–445, North-Holland, Amsterdam, the Netherlands.

**Summary.** The Levenberg–Morrison–Marquardt algorithm is compared to the dogleg method for sparse least-squares problems. Approximation schemes for the Jacobian are also considered.

- C. Reinsch (1971). Smoothing by spline functions II. *Numerische Mathematik*, **16**, 451–454.

- F. Rendl, R. J. Vanderbei, and H. Wolkowicz (1995). Max-min eigenvalue problems, primal-dual interior point algorithms, and trust region subproblems. *Optimization Methods and Software*, **5**(1), 1–16.

**Summary.** Two primal-dual interior-point algorithms are presented for maximizing the smallest eigenvalue of a symmetric matrix over diagonal perturbations. These algorithms prove to be simple, robust, and efficient. Both algorithms are based on transforming the problem to one with constraints over the cone of positive semidefinite matrices. One of the algorithms does this transformation through an intermediate transformation to a trust-region subproblem.

- F. Rendl and H. Wolkowicz (1997). A semidefinite framework for trust region subproblems with applications to large scale minimization. *Mathematical Programming*, **77**(2), 273–299.

**Summary.** A primal-dual pair of semidefinite programs provides a general framework for the theory and algorithms for the trust-region subproblem (TRS). This problem is a generalization of the minimum eigenvalue problem. The semidefinite framework is studied as an instance of semidefinite programming as well as a tool for viewing known algorithms and deriving new algorithms for TRS. In particular, a dual simplex-type method is studied that solves TRS as a parametric eigenvalue problem. This method uses the Lanczos algorithm for the smallest eigenvalue as a black box. Therefore, the essential cost of the algorithm is the matrix-vector multiplication, and thus sparsity can be exploited. A primal simplex-type method provides steps for the so-called hard case. Numerical tests for large sparse problems show that the cost of the algorithm is  $1 + \alpha(n)$  times the cost of finding a minimum eigenvalue using the Lanczos algorithm, where  $0 < \alpha(n) < 1$  is a fraction that decreases as the dimension increases.

- S. M. Robinson (1974). Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Mathematical Programming*, **7**(1), 1–16.

- S. M. Robinson (1983). Generalized equations. In A. Bachem, M. Grötschel, and B. Korte, eds., *Mathematical Programming: The State of the Art*, pp. 346–367, Springer-Verlag, Heidelberg, Berlin, New York.
- R. T. Rockafellar (1970). *Convex Analysis*. Princeton University Press, Princeton, NJ, USA.
- R. T. Rockafellar (1974). Augmented Lagrangian multiplier functions and duality in nonconvex programming. *SIAM Journal on Control and Optimization*, **12**(2), 268–285.
- R. T. Rockafellar (1976a). Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, **1**, 97–116.
- R. T. Rockafellar (1976b). Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, **14**, 877–898.
- R. T. Rockafellar (1983). Generalized subgradients in mathematical programming. In A. Bachem, M. Grötschel, and B. Korte, eds., *Mathematical Programming: The State of the Art*, pp. 368–390, Springer-Verlag, Heidelberg, Berlin, New York.
- J. F. Rodríguez, J. E. Renaud, and L. T. Watson (1998). Trust-region augmented Lagrangian methods for sequential response surface approximation and optimization. *Journal of Mechanical Design*, **120**(1), 58–66.

**Summary.** A common engineering practice is the use of approximation models in place of expensive computer simulations to drive a multidisciplinary design process based on nonlinear programming techniques. Well-established notions on trust-region methods are extended to manage the convergence of the general approximate problem where equality, inequality, and variable bound constraints are present. The primary concern is to manage the interaction between the optimization and the fidelity of the approximation models to ensure that the process converges to a solution of the original constrained design problem. This is achieved by using a trust-region model management strategy coupled with an augmented Lagrangian approach for constrained approximate optimization. An approximate optimization strategy is developed in which a cumulative response surface approximation of the augmented Lagrangian is sequentially optimized subject to a trust-region constraint. Results for several test problems are presented in which convergence to a Karush–Kuhn–Tucker point is observed.

- J. F. Rodríguez, J. E. Renaud, and L. T. Watson (1999). Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data. *Structural Optimization*, **15**(3–4), 141–156.

**Summary.** An augmented Lagrangian trust-region method is given that converges to a Karush–Kuhn–Tucker point for constrained optimization. The method behaves well on single-level optimization test problems. Applications include multidisciplinary design optimization test problems. It is shown that response surface approximations constructed from variable fidelity data generated during concurrent subspace optimizations can be effectively managed by the trust-region model management strategy.

- A. Roger, P. Terpolilli, and O. Gosselin (1988). Trust region methods for seismic inverse problems. In P. C. Sabatier, ed., *Proceedings of the XVIth Workshop on*

- Interdisciplinary Study of Inverse Problems: Some Topics on Inverse Problems*, pp. 93–103, World Scientific, Singapore.
- M. Rojas (1998). A large-scale trust-region approach to the regularization of discrete ill-posed problems. Ph.D. thesis, Rice University, Houston, TX, USA.
- Summary.** An algorithm based on the proposals by Sorensen (1997) and Santos and Sorensen (1995) is described for the solution of large-scale linear least-squares problems subject to a quadratic constraint. It is applied to regularize large discrete ill-posed problems. Numerical experiments on test cases and a real inverse interpolation problem illustrate the method.
- H. H. Rosenbrock (1960). An automatic method for finding the greatest or least value of a function. *Computer Journal*, **3**, 175–184.
- R. Roy and E. M. Sevick Muraca (1999). Truncated Newton's optimization scheme for absorption and fluorescence optical tomography. Part I: Theory and formulation. *Optics Express*, **4**(10), 353–371.
- Summary.** The development of noninvasive, biomedical optical imaging from time-dependent measurements of near-infrared (NIR) light propagation in tissues depends upon two crucial advances: (i) the instrumental tools to enable photon “time-of-flight” measurement within rapid and clinically realistic times, and (ii) the computational tools enabling the reconstruction of interior tissue optical property maps from exterior measurements of photon “time-of-flight” or photon migration. The image reconstruction algorithm is formulated as an optimization problem in which an interior map of tissue optical properties of absorption and fluorescence lifetime is reconstructed from synthetically generated exterior measurements of frequency-domain photon migration (FDPM). The inverse solution is accomplished using a truncated Newton's method with trust region to match synthetic fluorescence FDPM measurements with that predicted by the finite element prediction. The computational overhead and error associated with computing the gradient numerically is minimized upon using modified techniques of reverse automatic differentiation.
- M. Rudnicki (1994). Smoothing strategies in solving inverse electromagnetic problems. *International Journal of Applied Electromagnetics in Materials*, **4**(3), 239–264.
- Summary.** It is shown how the regularization parameter may be chosen when using the linear least-squares approach and the trust-region approach in the nonlinear least squares method for solving inverse and optimal design problems of electrical engineering. The zeroth-order stochastic method of optimization is also discussed, because it does not require restrictive a priori assumptions about convexity and smoothness. Some electromagnetic optimal design problems are solved by means of these techniques.
- Y. Saad (1991). *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, Manchester, England.
- Y. Saad (1996). *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, USA.
- E. W. Sachs and A. Sartenaer (2000). A class of augmented Lagrangian algorithms for infinite dimensional optimization with equality constraints. Technical Report (in preparation), Department of Mathematics, University of Namur, Belgium.
- E. W. Sachs, M. Schulze, and S. Fromme (1998). Neural networks—an application of numerical optimization in the financial markets. Presentation at the Optimization '98 Conference, Coimbra, Portugal.

**Summary.** The development of an efficient code to train neural networks and the use of this software to forecast some economic indicators is discussed. The proposed method exploits the underdetermined nature of the application and uses a Steihaug–Toint truncated conjugate gradient trust-region method. The software is applied to forecasting the German stock index, DAX. The success of the forecast is measured in the development of the value of a portfolio.

- S. J. Sadjadi and K. Ponnambalam (1999). Advances in trust region algorithms for constrained optimization. *Applied Numerical Mathematics*, **29**(3), 423–443.

**Summary.** A survey of recent advances in trust-region methods for constrained minimization is given. Different choices for the penalty function, Lagrange function, and expanded Lagrangian functions are compared. Some numerical results for an implementation of a recommended method on different test problems with various sizes are presented.

- N. Sagara and M. Fukushima (1991). A hybrid method for the nonlinear least-squares problem with simple bounds. *Journal of Computational and Applied Mathematics*, **36**(2), 149–157.

**Summary.** An interior trust-region method is presented for solving the nonlinear least-squares problem with lower and upper bounds on the variables. Convergence is established and the practical efficiency of the method is illustrated by numerical experiments.

- N. Sagara and M. Fukushima (1995). A hybrid method for solving the nonlinear least-squares problem with linear inequality constraints. *Journal of the Operations Research Society of Japan*, **38**(1), 55–69.

**Summary.** A trust-region method is presented for solving the nonlinear least-squares problem with linear inequality constraints. It is an adaptation of the method by Sagara and Fukushima (1991) to this more general case and enjoys similar properties.

- M. Sahba (1987). Globally convergent algorithm for nonlinearly constrained optimization problems. *Journal of Optimization Theory and Applications*, **52**(2), 291–309.

- D. E. Salane (1987). A continuation approach for solving large-residual nonlinear least squares problems. *SIAM Journal on Scientific and Statistical Computing*, **8**(4), 655–671.

**Summary.** A continuation method is used to develop a new trust-region framework for solving nonlinear least-squares problem and to motivate the direct selection of the trust-region parameter. It also provides a safeguard for trust-region methods and leads to a very robust algorithm. A class of algorithms based on the continuation method is presented. In addition, the implementation details for one member of the new class are examined. Convergence and descent properties of this algorithm are discussed. Numerical evidence is given showing that the new algorithms are competitive with existing trust-region algorithms.

- H. E. Salzer (1960). A note on the solution of quartic equations. *Mathematics of Computation*, **14**(71), 279–281.

- S. A. Santos and D. C. Sorensen (1995). A new matrix-free algorithm for the large-scale trust-region subproblem. Technical Report TR95-20, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A matrix-free algorithm for the large-scale trust-region subproblem is presented, which is claimed to improve upon the previous algorithms by introducing a unified iteration that naturally includes the hard case. The iteration is superlinearly convergent in all cases. Computational results are presented to illustrate its convergence properties and robustness.

- R. W. H. Sargent (1974). Reduced-gradient and projection methods for nonlinear programming. In P. E. Gill and W. Murray, eds., *Numerical Methods for Constrained Optimization*, pp. 149–174, Academic Press, London.
- R. W. H. Sargent and X. Zhang (1998). An interior-point algorithm for solving general variational inequalities and nonlinear programs. Presentation at the Optimization '98 Conference, Coimbra, Portugal.
- A. Sartenaer (1991). On some strategies for handling constraints in nonlinear optimization. Ph.D. thesis, Department of Mathematics, University of Namur, Belgium.
- A. Sartenaer (1993). Armijo-type condition for the determination of a generalized Cauchy point in trust region algorithms using exact or inexact projections on convex constraints. *Belgian Journal of Operations Research, Statistics and Computer Science*, **33**(4), 61–75.
- Summary.** An Armijo stepsize rule is proposed for the determination of a generalized Cauchy point and analysed for the trust-region methods proposed by Toint (1988) and Conn, Gould, Sartenaer, and Toint (1996b). It is proved under mild assumptions that both classes preserve their theoretical properties of global convergence and identification of the correct active set in a finite number of iterations. Numerical issues are discussed for both classes.
- A. Sartenaer (1995). A class of trust region methods for nonlinear network optimization problems. *SIAM Journal on Optimization*, **5**(2), 379–407.
- Summary.** The results of a series of tests upon a class of methods of trust-region type for solving the nonlinear network optimization problem are described. The trust-region technique considered is characterized by the use of the infinity norm and of inexact projections on the network constraints. The results are encouraging and show that this approach is particularly useful in solving large-scale nonlinear network optimization problems, especially when many bound constraints are expected to be active at the solution.
- A. Sartenaer (1997). Automatic determination of an initial trust region in nonlinear programming. *SIAM Journal on Scientific Computing*, **18**(6), 1788–1803.
- Summary.** A simple but efficient way to find a good initial trust-region radius in trust-region methods for nonlinear optimization is to monitor the agreement between the model and the objective function along the steepest-descent direction at the starting point. Further improvements for the starting point are also derived from the information gleaned during the initializing phase. Numerical results on a large set of problems show the impact the initial trust-region radius may have on trust-region methods' behaviour.
- Th. Sauer (1995). Computational aspects of multivariate polynomial interpolation. *Advances in Computational Mathematics*, **3**, 219–238.
- Th. Sauer and Y. Xu (1995). On multivariate Lagrange interpolation. *Mathematics of Computation*, **64**, 1147–1170.
- J. Schaepperle and E. Luder (1994). Optimization of distributed parameter systems with a combined statistical-deterministic method. In 1994 *IEEE International Symposium on Circuits and Systems*, Vol. 6, pp. 141–144, IEEE, New York, USA.
- Summary.** An optimization method is described for nonlinear systems with properties that depend on functions instead of discrete parameters. The method is applied to the design of

systems with spatially distributed parameters. The underlying mathematical problem of the calculus of variations is approximated by a finite-dimensional constrained nonlinear min-max problem. This is solved with a method that combines a deterministic algorithm for local optimization with a statistical method for global optimization. The former is based on linearization and linear programming with adaptive trust region, while the latter uses elements from genetic methods and pattern recognition. An example with a nonlinearly loaded nonuniform transmission line shows the capability of the algorithm to determine the unknown optimum function with high precision.

K. Schittkowski (1981). The nonlinear programming method of Wilson, Han and Powell with an augmented Lagrangian type line search function. *Numerische Mathematik*, **38**, 83–114.

S. Schleiff (1995). Parameterschätzung in nichtlinearen Modellen unter besonderer Berücksichtigung kritischer Punkte. Ph.D. thesis, Martin-Luther-Universität, Halle-Wittenberg, Germany.

T. Schlick (1993). Modified Cholesky factorizations for sparse preconditioners. *SIAM Journal on Scientific Computing*, **14**(2), 424–445.

R. B. Schnabel and E. Eskow (1991). A new modified Cholesky factorization. *SIAM Journal on Scientific Computing*, **11**(6), 1136–1158.

R. B. Schnabel and E. Eskow (1999). A revised modified Cholesky factorization. *SIAM Journal on Optimization*, **4**(9), 1064–1081.

R. B. Schnabel, J. E. Koontz, and B. E. Weiss (1985). A modular system of algorithms for unconstrained minimization. *ACM Transactions on Mathematical Software*, **11**(4), 419–440.

**Summary.** UNCMIN is a modular system of algorithms for optimization containing three different step selection strategies (linesearch, dogleg, and optimal step) that may be combined with either analytic or finite-difference gradient evaluation and with either analytic, finite-difference, or Broyden–Fletcher–Goldfarb–Shanno (BFGS) Hessian approximation. The results of a comparison of the three step selection strategies on the problems in Moré, Garbow, and Hillstrom (1981) are compared when using finite-difference gradients and Hessians or finite-difference gradients with BFGS Hessian approximations. A second package, REVMIN, that uses optimization algorithms identical to UNCMIN but obtains values of user-supplied functions by reverse communication is also described.

S. Scholtes and M. Stöhr (1999). Exact penalization of mathematical programs with equilibrium constraints. *SIAM Journal on Control and Optimization*, **37**(2), 617–652.

**Summary.** Theoretical and computational aspects of an exact penalization approach to mathematical programs with equilibrium constraints (MPEC) are studied. It is shown that a Mangasarian–Fromowitz-type condition ensures the existence of a stable local error bound at the root of a real-valued nonnegative piecewise smooth function. Specializing this to non-smooth formulations of equilibrium constraints, e.g., complementarity conditions or normal equations, then provides conditions that guarantee the existence of a nonsmooth exact penalty function for MPECs. A trust-region minimization method for a class of composite nonsmooth functions is then presented, which comprises exact penalty functions arising from MPECs. Global convergence is proved and a penalty update rule is described. A further specialization

- results in a sequential quadratic programming trust-region method for MPECs based on an  $\ell_1$  penalty function.
- H. Schramm and J. Zowe (1992). A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, **2**(1), 121–152.
- Summary.** A stable algorithm of the bundle family is obtained for solving nonsmooth minimization problems by adding some features of the trust-region philosophy to the bundle concept. The reliability and efficiency of the corresponding code is demonstrated on the standard academic examples and on some real-life problems.
- R. Schwencker, J. Eckmueller, H. Graeb, and K. Antreich (1999). Automating the sizing of analog CMOS circuits by consideration of structural constraints. In *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 323–327, IEEE Computer Society Press, Los Alamitos, CA, USA.
- Summary.** A method for the automatic sizing of analog integrated circuits is presented. Basic sizing rules, representing circuit knowledge, are set up before the sizing and are introduced as structural constraints into the sizing process. Systematic consideration of these structural constraints during the automatic sizing prevents pathologically sized circuits and speeds up the automatic sizing. The sizing is done with a sensitivity-based, iterative trust-region method.
- H. Schwetlick and V. Tiller (1989). Nonstandard scaling matrices for trust region Gauss–Newton methods. *SIAM Journal on Scientific and Statistical Computing*, **10**(4), 654–670.
- Summary.** Nonstandard scaling matrices are proposed for scaling the norm of the step in the solution of large nonlinear problems via the trust-region Gauss–Newton methods. The scaling matrices are rectangular, of full rank, and contain a block of the Jacobian matrix of the residual function. Three types of such matrices are investigated. The corresponding trust-region methods have qualitatively the same convergence properties as the standard method. Nonstandard scaling matrices are especially intended for solving large and structured problems such as orthogonal distance regression or surface fitting. Initial computational experience suggests that for such problems the proposed scaling sometimes implies a modest increase in the number of iterations but reduces overall computational cost.
- Ch. Sebudiandi (1992). Algorithmic developments in seismic tomography. Ph.D. thesis, Department of Mathematics, University of Namur, Belgium.
- Ch. Sebudiandi and Ph. L. Toint (1993). Nonlinear optimization for seismic travel time tomography. *Geophysical Journal International*, **115**, 929–940.
- Summary.** A nonlinear algorithm for seismic traveltimes analysis is presented based on large-scale nonlinear least-squares and trust-region methods. Numerical experience on synthetic data and on real borehole-to-borehole problems is presented. Results produced by the algorithms are compared with those of Ivansson for the Kråmåla experiment.
- J. Semple (1997). Optimality conditions and solution procedures for nondegenerate dual-response systems. *IIE Transactions*, **29**(9), 743–752.
- Summary.** The dual-response problem in the case where the response functions are nonconvex (nonconcave) quadratics and the independent variables satisfy a radial bound is investigated. Sufficient conditions for a global optimum are established and generalized to the multiresponse case. It is demonstrated that the sufficient conditions may not hold if the problem is “degenerate”. However, if the problem is nondegenerate, the sufficient conditions are necessarily satisfied by some stationary point. In this case, a specialized algorithm (DRSALG) locates

the global optimum in a finite number of steps. DRSALG also identifies the degenerate case and pinpoints the location where degeneracy occurs. The algorithm is easy to implement and is illustrated on a well-studied dual-response example from quality control.

- J. S. Shahabuddin (1996). Structured trust-region algorithms for the minimization of nonlinear functions. Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY, USA.

**Summary.** Trust-region algorithms are a popular and successful class of tools for the solution of nonlinear, nonconvex optimization problems. The basic trust-region algorithm is extended so that it takes advantage of partial separability to solve such large-scale problems in an efficient way. It aims at simplifying the proposal of Conn, Gould, Sartenaer, and Toint (1996b) in the case where no constraints are imposed in the problem. Three related approaches of “multiple” or “structured” trust regions are proposed. Convergence results are presented for them in the unconstrained case. Some computational results are discussed, in which the three approaches are compared.

- D. F. Shanno (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, **24**, 647–657.
- D. F. Shanno (1978). Conjugate gradient methods with inexact searches. *Mathematics of Operations Research*, **3**, 244–256.
- D. F. Shanno (1999). Topics in implementing an interior point method for nonconvex nonlinear programming. Presentation at the First Workshop on Nonlinear Optimization “Interior-Point and Filter Methods”, Coimbra, Portugal.
- Y. Sheffi (1985). *Urban Transportation Networks*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- T. Shiina (1999). Numerical solution technique for joint chance-constrained programming problem—an application to electric power capacity expansion. *Journal of the Operations Research Society of Japan*, **42**(2), 128–140.

**Summary.** A joint chance-constrained linear programming problem with random right-hand-side vector is considered. The deterministic equivalent of the joint chance-constraint is already known in the case that the right-hand-side vector is statistically independent. But if the right-hand-side vector is correlated, it is difficult to derive the deterministic equivalent of the joint chance-constraint. Two methods for calculating the joint chance-constraint are discussed. For the case of uncorrelated right-hand side, a direct method different from the usual deterministic equivalent is tried, while for the correlated right-hand-side case, numerical integration is applied. A chance-constrained programming problem is developed for electric power capacity expansion, where the error of forecasted electricity demand is defined by a random variable. This problem is solved numerically using a trust-region method and numerical integration, and results of computational experiments are presented.

- G. A. Shultz, R. B. Schnabel, and R. H. Byrd (1985). A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal on Numerical Analysis*, **22**(1), 47–67.

**Summary.** A general trust-region-based algorithm schema that includes an undefined step selection strategy is presented. General conditions on the step selection strategy under which limit points will satisfy first- and second-order necessary conditions are given. The algorithm schema is sufficiently broad to include linesearch methods as well. It is shown that a wide range

of step selection strategies satisfy the requirements of the convergence theory, and several algorithms that use second derivative information and achieve strong global convergence are proposed. These include an indefinite linesearch algorithm, several indefinite dogleg algorithms, and a modified “optimal-step” algorithm. An implementation of one such indefinite dogleg algorithm is proposed.

- E. M. Simantiraki and D. F. Shanno (1997). An infeasible-interior-point method for linear complementarity problems. In I. Duff and A. Watson, eds., *The State of the Art in Numerical Analysis*, pp. 339–362, Oxford University Press, Oxford, England.
- S. Smale (1983). On the average number of steps of the simplex method of linear programming. *Mathematical Programming*, **27**(3), 241–262.
- B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garbow, Y. Ikebe, V. C. Klema, and C. B. Moler (1976). *Matrix Eigensystem Routine—EISPACK Guide*. Springer-Verlag, Heidelberg, Berlin, New York.
- R. C. Smith and K. L. Bowers (1993). Sinc-Galerkin estimation of diffusivity in parabolic problems. *Inverse Problems*, **9**(1), 113–135.

**Summary.** A fully Sinc-Galerkin method for the numerical recovery of spatially varying diffusion coefficients in linear parabolic partial differential equations is presented. Because the parameter recovery problems are inherently ill-posed, an output error criterion in conjunction with Tikhonov regularization is used to formulate them as infinite-dimensional minimization problems. The forward problems are discretized with a Sinc basis in both the spatial and temporal domains, thus yielding an approximate solution that displays an exponential convergence rate and that is valid on the infinite time interval. The minimization problems are then solved via a quasi-Newton/trust-region algorithm. The L-curve technique for determining an appropriate value of the regularization parameter is briefly discussed, and numerical examples illustrate the applicability of the method both for problems with noise-free data as well as for those whose data contain white noise.

- D. C. Sorensen (1982a). Newton’s method with a model trust-region modification. *SIAM Journal on Numerical Analysis*, **19**(2), 409–426.

**Summary.** A trust-region Newton method for unconstrained minimization is presented and analysed. A thorough analysis of the locally constrained quadratic minimizations that arise as subproblems in the trust-region Newton iteration is given. Several promising alternatives are presented for solving these subproblems in ways that overcome certain theoretical difficulties exposed by the analysis. The explicit use of second-order information is justified by demonstrating that the iterates converge to a point that satisfies the second-order necessary conditions for minimization. With the exception of very pathological cases, this occurs whenever the algorithm is applied to problems with continuous second partial derivatives.

- D. C. Sorensen (1982b). Trust region methods for unconstrained optimization. In M. J. D. Powell, ed., *Nonlinear Optimization* 1981, pp. 29–39, Academic Press, London.

**Summary.** The basic trust-region approach to safeguarding Newton-like methods for unconstrained optimization is discussed.

- D. C. Sorensen (1997). Minimization of a large-scale quadratic function subject to a spherical constraint. *SIAM Journal on Optimization*, **7**(1), 141–161.

**Summary.** An algorithm is presented for solving the large-scale trust-region subproblem that requires a fixed-size limited storage proportional to the order of the quadratic and that relies only on matrix-vector products. The algorithm recasts the trust-region subproblem in terms of a parametrized eigenvalue problem and adjusts the parameter with a superlinearly convergent iteration to find the optimal solution from the eigenvector of the parametrized problem. Only the smallest eigenvalue and corresponding eigenvector of the parametrized problem need to be computed. The implicitly restarted Lanczos method is well suited to this subproblem.

- N. Stander and J. A. Snyman (1993). A new first-order interior feasible direction method for structural optimization. *International Journal for Numerical Methods in Engineering*, **36**(23), 4009–4025.

- T. Steihaug (1983). The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, **20**(3), 626–637.

**Summary.** It is shown that an approximate solution of the trust-region problem may be found by the preconditioned conjugate gradient method. This may be regarded as a generalized dogleg technique where asymptotically the inexact quasi-Newton step is taken. The resulting algorithm has the same properties as existing methods based on the dogleg strategy using an approximate Hessian.

- R. J. Stern and H. Wolkowicz (1994). Trust region problems and nonsymmetric eigenvalue perturbations. *SIAM Journal on Matrix Analysis and Applications*, **15**(3), 775–778.

**Summary.** A characterization is given for the spectrum of a symmetric matrix to remain real after a nonsymmetric sign-restricted border perturbation, including the case where the perturbation is skew-symmetric. The characterization is in terms of the stationary points of a quadratic function on the unit sphere. This yields interlacing relationships between the eigenvalues of the original matrix and those of the perturbed matrix. Applications include a characterization of matrices that are exponentially nonnegative with respect to the  $n$ -dimensional ice-cream cone, which leads to a decomposition theorem for such matrices. Results are obtained for nonsymmetric matrices regarding interlacing and majorization.

- R. J. Stern and H. Wolkowicz (1995). Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations. *SIAM Journal on Optimization*, **5**(2), 286–313.

**Summary.** The theory of trust-region subproblems is extended in two ways: (i) indefinite inner products in the quadratic constraint are allowed, and (ii) a two-sided (upper and lower bound) quadratic constraint is used. Characterizations of optimality are presented that have no gap between necessity and sufficiency. Conditions for the existence of solutions are given in terms of the definiteness of a matrix pencil. A simple dual program is introduced that involves the maximization of a strictly concave function on an interval. The dual program simplifies the theory and algorithms for trust-region subproblems. It also illustrates that they are implicit convex programming problems and thus explains why they are so tractable. The duality theory provides connections to eigenvalue perturbation theory. Trust-region subproblems with zero linear term in the objective function correspond to eigenvalue problems, and adding a linear term in the objective function is seen to correspond to a perturbed eigenvalue problem. Some eigenvalue interlacing results are presented.

- G. W. Stewart (1967). A modification of Davidon's minimization method to accept difference approximations of derivatives. *Journal of the ACM*, **14**(1), 72–83.

- J. Stoer (1983). Solution of large linear systems of equations by conjugate gradient type methods. In A. Bachem, M. Grötschel, and B. Korte, eds., *Mathematical*

- Programming: The State of the Art*, pp. 540–565, Springer-Verlag, Heidelberg, Berlin, New York.
- M. Stöhr (1999). Nonsmooth trust-region methods and their applications to mathematical programs with equilibrium constraints. Ph.D. thesis, Faculty of Mathematics, University of Karlsruhe, Germany.
- Summary.** A trust-region method for the solution of mathematical programs with equilibrium constraints (MPEC) is proposed and analysed. It makes use of exact penalty functions arising from the MPEC formulation. A variant of the algorithm by Scholtes and Stöhr (1999) that uses the concept of Cauchy point rather than requiring a model decrease proportional to that obtained at the global solution of the trust-region subproblem is discussed. Global convergence is proved and some numerical tests illustrate the method.
- Z. Strakoš (1991). On the real convergence rate of the conjugate gradient method. *Linear Algebra and Its Applications*, **154-156**, 535–549.
- G. Studer (1999). Risk measurement with maximum loss. *Mathematical Methods of Operations Research*, **50**(1), 121–134.
- Summary.** A worst-case portfolio selection problem is modelled as a trust-region subproblem. Ideas from trust-region theory are used to obtain an efficient solution.
- G. Studer and H.-J. Lüthi (1997). Maximum loss for risk measurement of portfolios. In U. Zimmermann, U. Derigs, W. Gaul, R. H. Mohring and K. P. Schuster, eds., *Operations Research Proceedings 1996: Selected Papers of the Symposium on Operations Research (SOR 96)*, pp. 386–391, Springer-Verlag, Heidelberg, Berlin, New York.
- Summary.** A brief review of the standard risk measure “value-at-risk” (VAR) is given and the concept of “maximum loss” (ML) for identifying the worst case in a given scenario space, a trust region, is introduced. A technique for efficiently calculating ML for quadratic functions is described; the algorithm is based on the Levenberg–Morrison–Marquardt theorem. The idea of the “maximum loss path” is presented. Repetitive calculation of ML for a growing trust region leads to a sequence of worst cases, which form a complete path. Similarly, the paths of “maximum profit” (MP) and “expected value” (EV) can be determined; their comparison permits judgements on the quality of portfolios. These concepts are applicable to nonquadratic portfolios by using “dynamic approximations”, which replace arbitrary profit and loss functions with a sequence of quadratic functions. The idea of “maximum loss distribution” is explained. The distributions of ML and MP can be obtained directly from the ML and MP paths, lead to lower and upper bounds of VAR, and allow statements about the spread of ML and MP.
- J. Sun (1993). A convergence proof for an affine-scaling algorithm for convex quadratic programming without nondegeneracy assumptions. *Mathematical Programming*, **60**(1), 69–79.
- J. Sun (1997). On piecewise quadratic Newton and trust-region problems. *Mathematical Programming*, **76**(3), 451–468.
- Summary.** Some algorithms for nonsmooth optimization require the solutions to certain piecewise quadratic programming subproblems. Two types of subproblems are considered. The first uses the minimization of a continuously differentiable and strictly convex piecewise quadratic function subject to linear equality constraints. A nonsmooth version of Newton’s method is globally and finitely convergent in this case. The second type involves the minimization of a possibly nonconvex and nondifferentiable piecewise quadratic function over a Euclidean ball. Characterizations of the global minimizer are studied under various conditions.

- J.-Q. Sun and K. Ruedenberg (1993). Quadratic steepest descent on potential energy surfaces. I. Basic formalism and quantitative assessment. *Journal of Chemical Physics*, **99**(7), 5257–5268.

**Summary.** A second-order algorithm is formulated for determining steepest-descent lines on potential energy surfaces, in which the stepsize is varied with the curvature and, if desired, readjusted by a trust-region assessment. Applications to the Gonzalez–Schlegel and the Muller–Brown surfaces show the method to behave well. Several measures are given for assessing the accuracy achieved without knowledge of the exact steepest-descent line. The optimal evaluation of the predicted gradient and curvature for dynamical applications is discussed.

- L. P. Sun (1996). A restricted trust region method with supermemory for unconstrained optimization. *Journal of Computational Mathematics*, **14**(3), 195–202.

**Summary.** A trust-region method for unconstrained optimization problems is presented, in which the descent direction is sought by using the trust-region steps within a restricted subspace. Because this subspace can be specified to include information about previous steps, the method is also related to supermemory descent methods. It is endowed with rapid convergence, as illustrated by numerical tests.

- W. Sun (1996). Optimization methods for nonquadratic model. *Asia-Pacific Journal of Operational Research*, **13**(1), 43–63.

- W. Sun and Y. Yuan (1998). A conic model trust region method for nonlinearly constrained optimization. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong.

**Summary.** Trust-region methods for constrained optimization using conic models are considered, and necessary and sufficient conditions are given for the solution of several equivalent formulations of the associated subproblems. Convergence properties of the resulting algorithm are established.

- M. Sunar and A. D. Belegundu (1991). Trust region methods for structural optimization using exact 2nd-order sensitivity. *International Journal for Numerical Methods in Engineering*, **32**(2), 275–293.

**Summary.** A trust-region method for structural optimization is constructed using analytical second-order sensitivity. The augmented Lagrangian  $\phi$  is minimized in this region subject to bounds. Evaluation of first and second derivatives of  $\phi$  by the adjoint method does not require derivations of individual (implicit) constraint functions, which makes the method economical. The algorithm is robust with respect to scaling, input parameters, and starting designs.

- W. A. Sutherland (1975). *Introduction to Metric and Topological Spaces*. Oxford University Press, Oxford, England.

- Y. Tanaka (1999). A trust region method for semi-infinite programming problems. *International Journal of Systems Science*, **30**(2), 199–204.

**Summary.** A trust-region SQP method, using second-order corrections, is applied to discretized semi-infinite programming problems, using an  $L_\infty$ -exact penalty function and  $\epsilon$ -most-active constraints. Preliminary computational experiments demonstrate the viability of this approach.

- Y. Tanaka, M. Fukushima, and T. Hasegawa (1987). Implementable  $l_\infty$  penalty-function method for semi-infinite optimization. *International Journal of Systems Science*, **18**(8), 1563–1568.

- Summary.** An implementable method for general nonlinear semi-infinite programming problems is described, which employs an exact  $L_\infty$  penalty function. Since this function is continuous even if the number of representative constraints changes, trust-region techniques may effectively be adopted to obtain global convergence. Numerical results are given to show the efficiency of the proposed algorithm.
- Y. Tanaka, M. Fukushima, and T. Ibaraki (1988). A globally convergent SQP method for semi-infinite nonlinear optimization. *Journal of Computational and Applied Mathematics*, **23**(2), 141–153.
- Summary.** A trust-region sequential quadratic programming method for semi-infinite programming is presented. The proposed algorithm employs the exact  $L_\infty$  penalty function and incorporates a scheme for estimating active constraints. It is proved to be globally convergent. The results of numerical experiments are reported.
- R. A. Tapia (1977). Diagonalized multiplier methods and quasi-Newton methods for constrained optimization. *Journal of Optimization Theory and Applications*, **22**, 135–194.
- C. Tappayuthpijarn and J. Jalali (1990). Loadflow solution by applying hybrid algorithm to the Newton-Raphson method. In *Proceedings of the American Power Conference, Illinois Institute of Technology, Chicago, IL, USA*, pp. 234–238.
- Summary.** A trust-region-based method is described for enforcing global convergence in solving power flow problems. A numerical comparison of the new method with classical approaches on a 10-bus system is given.
- M. Teboulle (1997). Convergence of proximal-like algorithms. *SIAM Journal on Optimization*, **6**(3), 617–625.
- D. J. Ternet (1994). A trust region algorithm for reduced Hessian successive quadratic programming. Ph.D. thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.
- Summary.** A sequential quadratic programming algorithm for solving nonlinear programming problems is reviewed, as is the role of linesearch and trust-region methods within this framework. A trust region is added to the current algorithm to increase the algorithm's robustness while maintaining its superlinear convergence properties. The benefits of combining a linesearch method with a trust-region method are explained. Test problems are described that motivate the need for the trust region.
- P. Terpolilli (1995). Trust region method in nonsmooth optimization. *Comptes Rendus de l'Académie des Sciences, Série Mathématique*, **321**(7), 945–948.
- Summary.** A framework for nonsmooth optimization is introduced. An algorithm using a trust-region strategy is considered and global convergence results are established. Particular attention is paid to the use of inexact local models. A convergence result is given in the situation where local models are computed by a numerical procedure.
- S. Thomas (1975). Sequential estimation techniques for quasi-Newton algorithms. Ph.D. thesis, Cornell University, Ithaca, NY, USA.
- R. Tibshirani (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, **58**(1), 267–288.

- W. F. Tinney and J. W. Walker (1967). Direct solution of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, **55**, 1801–1809.
- Ph. L. Toint (1977). On sparse and symmetric matrix updating subject to a linear equation. *Mathematics of Computation*, **31**(140), 954–961.
- Ph. L. Toint (1978). Some numerical result using a sparse matrix updating formula in unconstrained optimization. *Mathematics of Computation*, **32**(143), 839–851.
- Ph. L. Toint (1979). On the superlinear convergence of an algorithm for solving a sparse minimization problem. *SIAM Journal on Numerical Analysis*, **16**, 1036–1045.
- Ph. L. Toint (1980). Sparsity exploiting quasi-Newton methods for unconstrained optimization. In L. C. W. Dixon, E. Spedicato, and G. P. Szego, eds., *Nonlinear Optimization: Theory and Algorithms*, pp. 65–90, Birkhäuser, Boston.
- Ph. L. Toint (1981a). Convergence properties of a class of minimization algorithms that use a possibly unbounded sequence of quadratic approximations. Technical Report 81/1, Department of Mathematics, University of Namur, Belgium.
- Summary.** Global convergence results are established for a trust-region algorithm without assumptions on the norm of the Hessian approximations, but rather on the Rayleigh quotients of these approximations in certain directions. This allows for the case where Hessian approximations may become arbitrarily large provided they remain reasonable in these directions.
- Ph. L. Toint (1981b). Towards an efficient sparsity exploiting Newton method for minimization. In I. S. Duff, ed., *Sparse Matrices and Their Uses*, pp. 57–88, Academic Press, London.
- Summary.** Then-current algorithms for unconstrained minimization when the second derivative of the objective function is sparse are surveyed. Updating and estimation procedures are considered from the efficiency point of view. Special attention is given to the case where the Hessian has a band structure. A strategy for the choice of the step using truncated conjugate gradients is also discussed, and some numerical results on a specially designed test function are presented.
- Ph. L. Toint (1983a). User's guide to the routine PSPMIN for solving partially separable bounded optimization problems. Technical Report 83/1, Department of Mathematics, University of Namur, Belgium.
- Ph. L. Toint (1983b). VE08AD, a routine for partially separable optimization with bounded variables. *Harwell Subroutine Library*, User's manual, AERE Harwell, Oxfordshire, England.
- Ph. L. Toint (1987a). On large scale nonlinear least squares calculations. *SIAM Journal on Scientific and Statistical Computing*, **8**(3), 416–435.
- Summary.** The nonlinear model fitting problem is analysed, with special emphasis on the practical solution techniques when the number of parameters in the model is large. An extension of classical approaches to problems involving many variables is proposed that uses the concept of partially separable structures. An adaptable algorithm is discussed that chooses between various possible models of the objective function. Preliminary numerical experience

- shows that the solution of a large class of fitting problems involving several hundreds of non-linear parameters is possible at a reasonable cost.
- Ph. L. Toint (1987b). VE10AD, a routine for large scale nonlinear least squares. *Harwell Subroutine Library*, User's manual, AERE Harwell, Oxfordshire, England.
- Ph. L. Toint (1988). Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, **8**(2), 231–252.
- Summary.** A trust-region method for solving constrained optimization problems in Hilbert space is described. Global convergence results are derived without assuming convexity of the objective functional. It is also shown that convergence of the classical projected-gradient method can be viewed as a special case of this theory. An example is given that points out some difficulties that appear when using active-set strategies in infinite-dimensional spaces.
- Ph. L. Toint (1994). A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints: The complete numerical results. Technical Report 94/26, Department of Mathematics, University of Namur, Belgium.
- Ph. L. Toint (1996). An assessment of non-monotone linesearch techniques for unconstrained optimization. *SIAM Journal on Scientific and Statistical Computing*, **17**(3), 725–739.
- Ph. L. Toint (1997). A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. *Mathematical Programming*, **77**(1), 69–94.
- Summary.** Two trust-region methods for nonlinear optimization over convex feasible domains are presented. These methods are distinguished by the fact that they do not enforce strict monotonicity of the objective function values at successive iterates. The algorithms are proved to be convergent to critical points of the problem from any starting point. Extensive numerical experiments show that this approach is competitive with LANCELOT.
- X. Tong and X. Wang (1999). A trust-region algorithm for nonlinear network optimization problems. *Journal of Changsha University of Electric Power (Natural Science Edition)*, **14**(4), 299–302 (in Chinese).
- Summary.** A trust-region algorithm for a class of nonlinear network optimization problems is presented under general conditions. The global convergence result, which expresses that any accumulation point of the sequence generated by the algorithm is a Kuhn–Tucker point, is proved.
- X. Tong and S. Zhou (1999). A trust-region algorithm for nonlinear inequality constrained optimization. Technical Report July, Department of Mathematics, Hunan University, Changsha, China.
- Summary.** A trust-region algorithm for nonlinear optimization subject to nonlinear inequality constraints, based on an equivalent reformulation of the Karush–Kuhn–Tucker (KKT) conditions, is presented. Global convergence of the algorithm to a first-order KKT point is established under mild conditions on the trial step, and a local Q-quadratic convergence rate is attainable at a nondegenerate minimizer.
- P. Tseng (1999). A convergent infeasible interior-point trust-region method for constrained optimization. Technical Report, Department of Mathematics, University of Washington, Seattle, WA, USA.

**Summary.** A primal interior-point method that allows for infeasible points is presented for the inequality constrained nonlinear programming problem. The method uses a logarithmic barrier term for the slack variables and uses a trust region to find a step. The associated subproblem is solved exactly. Convergence to first-order critical points is proved, as is convergence to second-order ones under additional assumptions.

P. Tseng, N. Yamashita, and M. Fukushima (1996). Equivalence of complemetarity problems to differentiable minimization: A unified approach. *SIAM Journal on Optimization*, **6**(2), 446–460.

D. I. Tsioutsias and E. Mjolsness (1996). A multiscale attentional framework for relaxation neural networks. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, eds., *Advances in Neural Information Processing Systems. Proceedings of the 1995 Conference*, Vol. 8, pp. 631–639, MIT Press, Cambridge, MA, USA.

**Summary.** The optimization of neural networks governed by general objective functions is investigated. A novel framework is introduced for the solution of large-scale problems of this sort. It assumes little about the objective function and can be applied to general nonlinear, nonconvex functions, and objectives in thousand of variables are thus efficiently minimized by a combination of techniques such as deterministic annealing, multiscale optimization, attention mechanisms, and trust-region optimization methods.

T. Tsuchiya (1993). Global convergence of the affine scaling algorithm for primal degenerate strictly convex quadratic programming problems. *Annals of Operations Research*, **47**, 509–539.

H. W. Turnbull (1939). *Theory of Equations*. Oliver and Boyd, Edinburgh, London.

M. Ulbrich (1999). Non-monotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems. Technical Report TUM-M9906, Faculty of Mathematics, Technische Universität München, Germany.

**Summary.** A class of trust-region methods for bound-constrained semismooth systems of equations is developed, which is based on a simply constrained differentiable minimization reformulation. Global convergence results are proved that allow for nonmonotonicity of the function values at successive iterates. Trial steps are computed by a semismooth Newton-like method augmented by a projection onto the feasible set. Under a suitable condition and close to a regular solution, this technique turns into a projected Newton method, which converges locally Q-superlinearly or quadratically, depending on the quality of the approximate subdifferentials used. An application of this method to the solution of nonlinear mixed complementarity problems (MCPs) is then discussed, where the MCP is converted into a bound-constrained semismooth equation by means of an MCP function. A new class of MCP functions is introduced that is motivated by affine-scaling techniques for nonlinear programming. Numerical results for a subset of the MCPLIB problem collection illustrate the efficiency of this approach.

M. Ulbrich and S. Ulbrich (1997). Superlinear convergence of affine-scaling interior-point Newton methods for infinite-dimensional problems with pointwise bounds. Technical Report TR97-05, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A superlinearly convergent affine-scaling interior-point Newton method for infinite-dimensional problems with pointwise bounds in  $L^p$  space is analysed. The problem formulation is motivated by optimal control problems with  $L^p$  controls and pointwise control

- constraints. Adaptations are made to the proposal by Coleman and Li (1996a) for the infinite-dimensional setting. The main building block is a Newton-like iteration for an affine-scaling formulation of the Karush–Kuhn–Tucker condition. Using a pointwise projection, superlinear convergence under a weak strict complementarity condition and convergence with Q-rate  $> 1$  under a slightly stronger condition if a smoothing step is available are established. It is shown how the algorithm can be embedded in the class of globally convergent trust-region interior-point methods of Ulbrich, Ulbrich, and Heinkenschloss (1999). Numerical results for the control of a heating process confirm the theoretical findings.
- M. Ulbrich, S. Ulbrich, and M. Heinkenschloss (1999). Global convergence of trust-region interior-point algorithms for infinite dimensional nonconvex minimization subject to pointwise bounds. *SIAM Journal on Control and Optimization*, **37**(3), 731–764.
- Summary.** Interior-point trust-region algorithms for infinite-dimensional nonlinear optimization subject to pointwise bounds in  $L^p$  Banach spaces,  $2 \leq p \leq \infty$ , are analysed. The problem formulation is motivated by optimal control problems, with  $L^p$ -controlled and pointwise control constraints. The interior-point trust-region algorithms are generalizations of those introduced by Coleman and Li (1996a) for finite-dimensional problems. Many of the generalizations lead to a better understanding of the methods and to considerable improvements in their performance. All first- and second-order global convergence results known in the finite-dimensional setting are extended to the infinite-dimensional framework.
- S. Ulbrich and M. Ulbrich (1999). Nonmonotone trust region methods for nonlinear equality constrained optimization without a penalty function. Presentation at the First Workshop on Nonlinear Optimization “Interior-Point and Filter Methods”, Coimbra, Portugal.
- Summary.** A class of nonmonotone trust-region methods for nonlinear equality constrained optimization problems is proposed, where each step is composed of a quasi-normal and a tangential step. Both steps are required to satisfy a fraction of the Cauchy decrease condition for their respective trust-region subproblems, and the mechanism for accepting them combines nonmonotone decrease conditions on the constraint violation and/or the objective function. Preliminary numerical results show considerable promise.
- T. Urban, A. L. Tits, and C. L. Lawrence (1998). A primal-dual interior-point method for nonconvex optimization with multiple logarithmic barrier parameters and with strong convergence properties. Technical Report TR 98-27, Electrical Engineering and the Institute for Systems Research, University of Maryland, College Park, MD, USA.
- C. Van de Panne and A. Whinston (1969). The symmetric formulation of the simplex method for quadratic programming. *Econometrica*, **37**, 507–527.
- S. Van Huffel, editor (1997). *Recent Advances in Total Least-Squares Techniques and Error-in-Variables Modeling*, SIAM, Philadelphia, USA.
- S. Van Huffel and J. Vandewalle (1991). *The Total Least-Squares Problem: Computational Aspects and Analysis*. Frontiers in Applied Mathematics 9, SIAM, Philadelphia, USA.
- L. Vandenberghe and S. Boyd (1996). Semidefinite programming. *SIAM Review*, **38**(1), 49–95.

- R. J. Vanderbei (1994). LOQO: An interior point code for quadratic programming. Technical Report SOR 94-15, Program in Statistics and Operations Research, Princeton University, Princeton, NJ, USA.
- R. J. Vanderbei and D. F. Shanno (1997). An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR 97-21, Program in Statistics and Operations Research, Princeton University, Princeton, NJ, USA.
- J. S. Vandergraft (1985). Efficient optimization methods for maximum likelihood parameter estimation. In *Proceedings of the 24th IEEE Conference on Decision and Control*, Vol. 3, pp. 1906–1909, IEEE, New York, USA.
- Summary.** It is shown how the most successful of the quasi-Newton methods and more conventional methods such as scoring can be combined with a trust-region technique to produce numerical algorithms that are ideally suited to maximum-likelihood parameter estimation. Specific properties of these algorithms include a superlinear rate of convergence and the ability to handle parameter constraints easily and efficiently.
- A. Vardi (1985). A trust region algorithm for equality constrained minimization: Convergence properties and implementation. *SIAM Journal on Numerical Analysis*, **22**(3), 575–591.
- Summary.** A trust-region strategy for equality constrained minimization is developed. This algorithm is analysed and global as well as local superlinear convergence theorems are proved. It is demonstrated how to implement this algorithm in a numerically stable way.
- A. Vardi (1992). New minimax algorithms. *Journal of Optimization Theory and Applications*, **75**(3), 613–634.
- Summary.** An algorithm for nonlinear min-max is described that reformulates the min-max function as a set of inequality constraints and uses an active-set, trust-region method that exploits the structure of the resulting problem. Numerical results are presented.
- S. A. Vavasis (1992a). Approximation algorithms for indefinite quadratic programming. *Mathematical Programming*, **57**(2), 279–311.
- S. A. Vavasis (1992b). *Nonlinear Optimization: Complexity Issues*. International Series of Monographs on Computer Science, Oxford University Press, Oxford, England.
- S. A. Vavasis and R. Zippel (1990). Proving polynomial-time for sphere-constrained quadratic programming. Technical Report TR 90-1182, Department of Computer Science, Cornell University, Ithaca, NY, USA.
- Summary.** Ye (1989) and Karmarkar (1989, unpublished manuscript) have proposed similar algorithms for minimizing a nonconvex quadratic function on a sphere. Estimates are derived for the convergence of the algorithm based on bounds for separation of roots of polynomials. These bounds prove that the underlying decision problem is polynomial time in the Turing machine sense.
- L. N. Vicente (1995). Trust-region interior-point algorithms for a class of nonlinear programming problems. Ph.D. thesis and Report TR96-05, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A family of trust-region interior-point (TRIP) reduced sequential quadratic programming (SQP) algorithms, for the solution of minimization problems with nonlinear equality constraints and simple bounds, is introduced and analysed. These problems appear in control, design, parameter identification, and inversion. In particular, they often arise in the discretization of optimal control problems. The TRIP reduced SQP algorithms treat states and controls as independent variables; they do not rely on matrix factorizations of the linearized constraints, but use solutions of the linearized state and adjoint equations. These algorithms result from a combination of a reduced SQP algorithm, a trust-region globalization, and a primal-dual affine-scaling interior-point method. They converge globally and quadratically to points satisfying first- and second-order necessary optimality conditions. The algorithms and convergence results reduce to those of Coleman and Li (1996a) for box-constrained optimization. An inexact analysis is presented to provide a practical way of controlling residuals of linear systems and directional derivatives. Numerical experiments for two nonlinear optimal control problems illustrate the robustness and effectiveness of these algorithms. A specialized analysis for equality-constrained problems shows that they do not require the computation of normal components for the step and an orthogonal basis for the null-space of the Jacobian of the equality constraints.

- L. N. Vicente (1996). A comparison between line searches and trust regions for nonlinear optimization. *Investigação Operacional*, **16**(2), 173–179.

**Summary.** It is claimed that the trust-region technique has an appropriate built-in regularization of ill-conditioned second-order approximation that is lacking in linesearch methods. To justify this claim, the trust-region technique is forced to act like a linesearch by always choosing the step along the quasi-Newton direction. Global convergence to a critical point is obtained so long as the condition number of the second-order approximation is uniformly bounded, a condition required in linesearches but not in trust regions.

- C. R. Vogel (1990). A constrained least squares regularization method for nonlinear ill-posed problems. *SIAM Journal on Control and Optimization*, **28**(1), 34–49.

**Summary.** A trust-region method is applied for regularizing ill-posed, nonlinear Hilbert space operator equations. The subproblem is reformulated as a nonlinear complementarity problem and solved using Newton's method. The method of generalized cross validation is used to pick the regularization parameter when random error is present in the discrete data. The method is applied to find approximate solutions to a severely ill-posed nonlinear first kind integral equation arising in geophysics.

- G. A. Watson (1980). *Approximation Theory and Numerical Methods*. Wiley, Chichester, England.

- L. T. Watson, S. C. Billups, and A. P. Morgan (1987). HOMPACK: A suite of codes for globally convergent homotopy algorithms. *ACM Transactions on Mathematical Software*, **13**(3), 281–310.

- L. T. Watson, M. P. Kamat, and M. H. Reaser (1985). A robust hybrid algorithm for computing multiple equilibrium solutions. *Engineering Computations*, **2**, 30–34.

**Summary.** A hybrid method is described that combines the efficiency of a quasi-Newton method capable of locating stable and unstable equilibrium configurations with a robust homotopy method that is capable of tracking equilibrium paths with turning points while exploiting symmetry and sparsity of the Jacobian matrices. The quasi-Newton method uses a double-dogleg trust-region strategy. Numerical results are presented for a shallow-arch problem.

- C. Weihs, G. Calzolari, and L. Panattoni (1987). The behavior of trust-region methods in FIML-estimation. *Computing*, **38**(2), 89–100.

**Summary.** The reliability of numerical algorithms for full information, maximum likelihood estimation in nonlinear econometric models is explored by a Monte-Carlo study. Suitable Hessian approximations within trust-region algorithms are compared with regard to their robustness and their global and local convergence speed. With respect to robustness and global convergence speed, the crude generalized least-squares-type Hessian approximations perform best, efficiently exploiting the special structure of the likelihood function. But, if the speed of local convergence is of utmost convergence, general-purpose techniques are strongly superior. Some appropriate mixture of these two types of approximations is recommended.

- J. H. Wilkinson (1963). *Rounding Errors in Algebraic Processes*. Her Majesty's Stationery Office, London.

- J. H. Wilkinson (1965). *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, England.

- J. H. Wilkinson (1968). A priori error analysis of algebraic processes. In I. G. Petrovsky, ed., *Proceedings of the International Congress of Mathematicians*, pp. 629–640, Mir Publishers, Moscow, USSR.

- J. W. J. Williams (1964). Algorithm 232, Heapsort. *Communications of the ACM*, **7**, 347–348.

- K. A. Williamson (1990). A robust trust region algorithm for nonlinear programming. Technical Report TR90-22, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA.

**Summary.** A variant of the trust-region algorithm by Celis, Dennis, and Tapia (1985) for general nonlinear programming is developed that uses a two-dimensional subproblem. This subproblem consists of globally minimizing a possibly nonconvex quadratic subject to two quadratic constraints in two dimensions. A detailed study of this subproblem is supplied for a number of special cases. Preliminary numerical experiments illustrate the robustness of the resulting algorithm.

- R. B. Wilson (1963). A simplicial algorithm for concave programming. Ph.D. thesis, Harvard University, Cambridge, MA, USA.

- D. Winfield (1969). Function and functional optimization by interpolation in data tables. Ph.D. thesis, Harvard University, Cambridge, MA, USA.

- D. Winfield (1973). Function minimization by interpolation in a data table. *Journal of the Institute of Mathematics and Its Applications*, **12**, 339–347.

**Summary.** A method is described for unconstrained function minimization using function values and no derivatives. A quadratic model of the function is formed by interpolation to points in a table of function values. The quadratic model (not necessarily positive definite) is minimized over a constraining region of validity to locate the next trial point. The points of interpolation are chosen from a data table containing function values at an initial grid and at subsequent trial points. The method is efficient in its use of function evaluations but expensive in computation required to choose new trial points.

- J. Wloka (1987). *Partial Differential Equations*. Cambridge University Press, Cambridge, England.
- P. Wolfe (1959). The Simplex method for quadratic programming. *Econometrica*, **27**, 382–398.
- R. S. Womersley (1982). Optimality conditions for piecewise smooth functions. *Mathematical Programming Studies*, **17**, 13–27.
- R. S. Womersley (1985). Local properties of algorithms for minimizing nonsmooth composite functions. *Mathematical Programming*, **32**(1), 69–89.
- T. H. Wonnacott and R. J. Wonnacott (1990). *Introductory Statistics*, fifth ed. Wiley, Chichester, England.
- M. H. Wright (1976). Numerical methods for nonlinearly constrained optimization. Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, USA.
- M. H. Wright (1992). Interior methods for constrained optimization. *Acta Numerica*, **1**, 341–407.
- M. H. Wright (1995). Why a pure primal Newton barrier step may be infeasible. *SIAM Journal on Optimization*, **5**(1), 1–12.
- M. H. Wright (1998). The interior-point revolution in constrained optimization. In R. De Leone, A. Murli, P. M. Pardalos, and G. Toraldo, eds., *High Performance Algorithms and Software in Nonlinear Optimization*, pp. 359–381, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- M. H. Wright (1999). Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM Journal on Optimization*, **9**(1), 84–111.
- S. J. Wright (1987). Local properties of inexact methods for minimizing nonsmooth composite functions. *Mathematical Programming*, **37**(2), 232–252.
- S. J. Wright (1989a). Convergence of SQP-like methods for constrained optimization. *SIAM Journal on Control and Optimization*, **27**(1), 13–26.
- S. J. Wright (1989b). An inexact algorithm for composite nondifferentiable optimization. *Mathematical Programming*, **44**(2), 221–234.
- Summary.** An inexact version of Fletcher's (1987a) *QL* trust-region algorithm with second-order corrections for minimizing composite nonsmooth functions, which retains the global and local convergence properties of the original version, is given. It is shown how the inexact method can be implemented for exact penalty functions arising from nonlinear programming problems, as well as problems of nonlinear  $\ell_1$  and  $\ell_\infty$  approximation.
- S. J. Wright (1990). Convergence of an inexact algorithm for composite nonsmooth optimization. *IMA Journal of Numerical Analysis*, **10**(3), 299–321.

**Summary.** An inexact version of Fletcher's (1982b) second-order correction algorithm for minimizing composite nondifferentiable functions is described. A test is suggested that allows global convergence to be proved without the assumption that a global minimum of the model function is found at each iteration. Implementable criteria for accepting inexact solutions of the subproblem, while retaining local convergence properties, are given.

S. J. Wright (1997). *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, USA.

S. J. Wright (1998). Effects of finite-precision arithmetic on interior-point methods for nonlinear programming. Technical Report MCS-P705-0198, Argonne National Laboratory, Argonne, IL, USA.

S. J. Wright (1999). Superlinear convergence of a stabilized SQP method to a degenerate solution. *Computational Optimization and Applications*, **11**(3), 253–275.

S. J. Wright and J. N. Holt (1985). Algorithms for nonlinear least squares with linear inequality constraints. *SIAM Journal on Scientific and Statistical Computing*, **6**(4), 1033–1048.

**Summary.** Two algorithms for solving nonlinear least squares problems with general linear inequality constraints are described. At each step, the problem is reduced to a trust-region constrained linear least-squares problem in the subspace defined by the active constraints, which is solved using the Levenberg–Morrison–Marquardt method. The desirability of leaving an active constraint is evaluated at each step, using a different technique for each of the two algorithms. Comparisons of the relative performance of the two algorithms on small problems and on a larger exponential data-fitting problem are presented.

S. J. Wright and D. Orban (1999). Properties of the log-barrier function for degenerate nonlinear programs. Technical Report TR/PA/99/36, CERFACS, Toulouse, France.

Y. Xiao (1996). Non-monotone algorithms in optimization and their applications. Ph.D. thesis, Monash University, Clayton, Australia.

Y. Xiao and E. K. W. Chu (1995). Nonmonotone trust region methods. Technical Report 95/17, Monash University, Clayton, Australia.

**Summary.** Two nonmonotone trust-region methods are developed and assessed based on extensive numerical experiments with CUTE. Strategies for automatic adjustment of parameters are discussed, which enable switching between nonmonotone and monotone algorithms at different stages of calculation according to the intermediate information obtained. Numerical results show that these strategies improve the efficiency of nonmonotone algorithms. Global convergence of the algorithms is proved, and further modifications are discussed.

Y. Xiao and F. Zhou (1992). Nonmonotone trust region methods with curvilinear path in unconstrained optimization. *Computing*, **48**(3–4), 303–317.

**Summary.** A nonmonotone trust-region algorithm is proposed for unconstrained optimization, whose convergence properties are similar to those of the monotone versions. Numerical experiments show the potential benefits of the approach.

C. Xu and J. Zhang (1995). An active set method for general linear  $\ell_1$  problem subject to box constraints. *Optimization*, **34**(1), 67–80.

- Summary.** An active set algorithm is presented for the solution of a general  $\ell_1$  linear problem with simple bound constraints on its variables. The implication for trust-region subproblems is discussed.
- C. Xu and J. Zhang (1999). A scaled optimal path trust region algorithm. *Journal of Optimization Theory and Applications*, **102**(1), 127–146.
- Summary.** A scaled optimal path trust-region algorithm is proposed, which finds a solution of the subproblem in full-dimensional space by just one Bunch and Parlett (1971) factorization and by using the resulting unit lower triangular factor to scale the variables. The resulting algorithm has strong convergence properties. Computational results show that this algorithm is robust and effective.
- H. Yabe and H. Yamashita (1997). Q-superlinear convergence of primal-dual interior point quasi-Newton methods for constrained optimization. *Journal of the Operations Research Society of Japan*, **40**(3), 415–436.
- E. Yamakawa, M. Fukushima, and T. Ibaraki (1989). An efficient trust region algorithm for minimizing nondifferentiable composite functions. *SIAM Journal on Scientific and Statistical Computing*, **10**(3), 562–580.
- Summary.** A trust-region method for solving the problem of minimizing  $\phi(x) = f(x) + h(c(x))$ , where  $f$  and  $c$  are smooth functions and  $h$  is a polyhedral convex function, is given. The algorithm is an adaptation of the sequential quadratic programming method by Fletcher (1982a) and makes use of second-order approximations to both  $f$  and  $c$  in order to avoid the Maratos effect. Global and quadratic convergence is proved. Numerical results illustrate the effectiveness of the algorithm.
- H. Yamashita (1982). A globally convergent constrained quasi-Newton method with an augmented Lagrangian type penalty-function. *Mathematical Programming*, **23**(1), 75–86.
- H. Yamashita and H. Yabe (1996a). Nonmonotone SQP methods with global and superlinear convergence properties. Technical Report, Mathematical Systems, Inc., Sinjuku-ku, Tokyo, Japan.
- H. Yamashita and H. Yabe (1996b). Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization. *Mathematical Programming, Series A*, **75**(3), 377–397.
- H. Yamashita, H. Yabe, and T. Tanabe (1997). A globally and superlinearly convergent primal-dual point trust region method for large scale constrained optimization. Technical Report, Mathematical Systems, Inc., Sinjuku-ku, Tokyo, Japan.
- Summary.** A primal-dual interior-point method is proposed, where the barrier function is minimized by a trust-region method that uses second derivatives. Superlinear convergence is proved. A nonmonotone strategy is adopted to avoid the Maratos effect as in Yamashita and Yabe (1996a). Results are reported for a variety of problems given by Hock and Schittkowski from the CUTE test set.
- N. Yamashita and M. Fukushima (1995). On stationary points of the implicit Lagrangian for nonlinear complementarity problems. *Journal of Optimization Theory and Applications*, **84**(3), 653–663.

- B. Yang, K. Zhang, and Z. You (1996). A successive quadratic programming method that uses new corrections for search directions. *Journal of Computational Mathematics*, **71**(1), 15–31.
- E. K. Yang and J. W. Tolle (1991). A class of methods for solving large, convex quadratic programs subject to box constraints. *Mathematical Programming*, **51**(2), 223–228.
- Y. Yang, D. Li, and S. Zhou (1998). A trust region method for a semismooth reformulation to variational inequality problems. Technical Report, May 15, Department of Applied Mathematics, Hunan University, Changsha, China.
- Summary.** A trust-region method is proposed for solving general nonlinearly constrained variational inequality problems. It is based on the semismooth reformulation of the first-order optimality conditions, allows for inexact solution of subproblems, and is globally convergent even in the nonmonotone case. Its rate of convergence is Q-superlinear or Q-quadratic, even in the absence of strict complementarity.
- Y. Ye (1989). An extension of Kamarkar's algorithm and the trust region method for quadratic-programming. In N. Megiddo, ed., *Progress in Mathematical Programming*, pp. 49–63, Springer-Verlag, Heidelberg, Berlin, New York.
- Summary.** An extension of Karmarkar's (1984) algorithm and the trust-region method is developed for solving quadratic programming problems. It is based on the affine-scaling technique, followed by optimization over a trust ellipsoidal region, and creates a sequence of interior feasible points that converge to the optimal solution. Computational results suggest its potential usefulness.
- Y. Ye (1992). On an affine scaling algorithm for nonconvex quadratic programming. *Mathematical Programming*, **56**, 285–300.
- Summary.** The use of interior algorithms, especially the affine-scaling algorithm, to solve nonconvex—definite or negative definite—quadratic programming (QP) problems is investigated. Although the nonconvex QP with a polytope constraint is a “hard” problem, it is shown that the problem with an ellipsoidal constraint is “easy”. When the “hard” QP is solved by successively solving the “easy” QP, the sequence of points monotonically converges to a feasible point satisfying both the first- and the second-order optimality conditions.
- Y. Ye (1997). Approximating quadratic programming with bound constraints. Working Paper, Department of Management Sciences, University of Iowa, Ames, IA, USA.
- Summary.** The problem of approximating the global maximum of a quadratic program with  $n$  variables subject to bound constraints is considered. Based on the results of Goemans and Williamson (1996) and Nesterov (1998), it is shown that a  $4/7$  approximate solution can be obtained in polynomial time.
- Y. Ye and E. Tse (1989). An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, **44**(2), 157–179.
- H. Yin and J. Han (1998). A new interior-point trust-region algorithm for nonlinear minimization problems with simple bound constraints. Presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong.

**Summary.** In the method of Coleman and Li (1996a), feasibility of the trial step is obtained by backtracking from a possibly infeasible step into the interior of the feasible region. A variant of this method is discussed in which the initial trial step is not allowed to be infeasible, therefore avoiding the need of backtracking.

K. Yosida (1970). *Functional Analysis*. Springer-Verlag, Heidelberg, Berlin, New York.

Y. Yuan (1983). Global convergence of trust region algorithms for nonsmooth optimization. Technical Report DAMTP/NA13, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, England.

**Summary.** This is a preliminary version of Yuan (1985a).

Y. Yuan (1984). An example of only linear convergence of trust region algorithms for nonsmooth optimization. *IMA Journal of Numerical Analysis*, **4**(3), 327–335.

**Summary.** An example is constructed where, at every iteration of a trust-region method for nonsmooth optimization the trust-region bound is active and the rate of convergence is only linear, even though strict complementarity and second-order sufficiency conditions hold.

Y. Yuan (1985a). Conditions for convergence of trust region algorithms for nonsmooth optimization. *Mathematical Programming*, **31**(2), 220–228.

**Summary.** Properties of trust-region algorithms for nonsmooth optimization are discussed. The problem is expressed as the minimization of a function  $h(f(x))$  where  $h(\cdot)$  is convex and where  $f$  is a continuously differentiable mapping from  $\mathbb{R}^n$  to  $\mathbb{R}^n$ . It is shown that the results of Powell (1975, 1984) hold for nonsmooth optimization.

Y. Yuan (1985b). On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Mathematical Programming*, **31**(3), 269–285.

**Summary.** It is proved that the second-order correction trust-region algorithm of Fletcher (1982b) ensures superlinear convergence if some mild conditions are satisfied.

Y. Yuan (1990). On a subproblem of trust region algorithms for constrained optimization. *Mathematical Programming*, **47**(1), 53–63.

**Summary.** A subproblem that arises in some trust-region algorithms for equality constrained optimization is studied. It is the minimization of a general quadratic function with two special quadratic constraints. Properties of such subproblems are given. It is proved that the Hessian of the Lagrangian has at most one negative eigenvalue, and an example is presented to show that the Hessian may have a negative eigenvalue when one constraint is inactive at the solution.

Y. Yuan (1991). A dual algorithm for minimizing a quadratic function with two quadratic constraints. *Journal of Computational Mathematics*, **9**(4), 348–359.

**Summary.** A dual globally and superlinearly convergent algorithm is proposed for minimizing a convex quadratic objective subject to two quadratic constraints. Numerical examples are provided.

Y. Yuan (1993). A new trust-region algorithm for nonlinear optimization. In D. Bainov and V. Covachev, eds., *Proceedings of the First International Colloquium on Numerical Analysis*, pp. 141–152, VSP, Zeist, the Netherlands.

**Summary.** Trust-region algorithms are a class of numerical algorithms for optimization. A trust-region algorithm for general nonlinear constrained optimization problems is presented. The algorithm is based on the  $\ell_\infty$  exact penalty function. Under very mild conditions, global convergence results for the algorithm are given.

- Y. Yuan (1994a). Nonlinear programming: Trust region algorithms. In S. T. Xiao and F. Wu, eds., *Proceedings of Chinese SIAM Annual Meeting*, pp. 83–97, Tsinghua University Press, Beijing, China.

**Summary.** A brief survey of trust-region methods for constrained nonlinear optimization is presented.

- Y. Yuan (1994b). On the convergence of trust region algorithms. *Mathematica Numerica Sinica*, **16**(3), 333–346 (in Chinese).

**Summary.** Trust-region algorithms for nonlinear optimization and their convergence properties are discussed. A generic descent trial step is defined and is used to obtain a unified proof for global convergence of such algorithms.

- Y. Yuan (1994c). Trust region algorithms for constrained optimization. In J. Cui, Z. Shi, and D. Wang, eds., *Proceedings of Conference on Scientific and Engineering Computing for Young Chinese Scientists*, pp. 105–110, National Defence Industry Press, Beijing, China.

**Summary.** A survey of trust-region methods for nonlinear optimization is given, with emphasis on globally and locally superlinear convergence properties.

- Y. Yuan (1994d). Trust region algorithms for nonlinear programming. In Z. C. Shi, ed., *Contemporary Mathematics*, Vol. 163, pp. 205–225, American Mathematical Society, Providence, RI, USA.

**Summary.** A review of the many results in the domain of trust-region methods for nonlinear optimization and the solution of nonlinear systems of algebraic equations is presented.

- Y. Yuan (1995). On the convergence of a new trust region algorithm. *Numerische Mathematik*, **70**(4), 515–539.

**Summary.** A trust-region algorithm is presented for general nonlinear constrained problems, based on the  $\ell_\infty$  exact penalty function. Under very mild conditions, global and local convergence results for the algorithm are given. It is shown that the penalty parameter generated by the algorithm will eventually be not less than the  $\ell_1$  norm of the Lagrange multipliers at the accumulation point. It is proved that the method is equivalent to the sequential quadratic programming method for all large  $k$ , hence superlinearly convergent results for the sequential quadratic programming method can be applied. Numerical results are reported.

- Y. Yuan (1996). A short note on the Duff-Nocedal-Reid algorithm. *SEA Bulletin of Mathematics*, **20**(3), 137–144.

**Summary.** An example is given to show that the algorithm of Duff, Nocedal, and Reid (1987) for nonlinear equations may converge to a nonoptimal solution. It is also shown that a slight modification can ensure the global convergence of the algorithm.

- Y. Yuan (1997). Some properties of a trust region subproblem. Presentation at the XVIth International Symposium on Mathematical Programming, Lausanne, Switzerland.

**Summary.** It is shown that the model reduction obtained by applying the Steihaug–Toint algorithm on a convex quadratic model in two dimensions is at least half of that obtained by the exact minimizer of the model within the trust region.

- Y. Yuan (1998a). An example of non-convergence of trust region algorithms. In Y. Yuan, ed., *Advances in Nonlinear Programming*, pp. 205–218, Kluwer Academic Publishers, Dordrecht, the Netherlands.

- Summary.** An example is constructed which shows that it can happen that for a class of trust-region algorithms that do not require sufficient reductions the whole sequence need not converge. In the example, only one accumulation point is a stationary point while all other accumulation points are nonstationary.
- Y. Yuan (1998b). Matrix computation problems in trust region algorithms for optimization. In Q. C. Zeng, T. Q. Li, Z. S. Xue, and Q. S. Cheng, eds., *Proceedings of the 5th CSIAM Annual Meeting*, pp. 54–64, Tsinghua University Press, Beijing, China.
- Summary.** The linear algebra aspects of the solution methods for various trust-region subproblems are reviewed.
- Y. Yuan (1998c). Optimality conditions for the Celis-Dennis-Tapia subproblems. Presentation at the Optimization ‘98 Conference, Coimbra, Portugal.
- Summary.** Easily verifiable necessary and sufficient optimality conditions are given for local solutions of the Celis–Dennis–Tapia trust-region subproblem. If the suproblem has no global solution for which the Hessian of the Lagrangian is positive semidefinite, the Hessian at any local solution has at least one negative eigenvalue. Some other characteristics of local solutions are also given. The gap between necessary and sufficient conditions is discussed.
- Y. Yuan (1998d). Trust region algorithms for nonlinear equations. *Information*, **1**, 7–20.
- Summary.** The problem of solving nonlinear equations  $F(x) = 0$ , where  $F(x)$  from  $\mathbb{R}^n$  to  $\mathbb{R}^m$  is continuously differentiable, is considered. A class of general trust-region algorithms for solving nonlinear equations by minimizing a given norm  $\|F(x)\|$  is studied. The trust-region algorithm for nonlinear equations can be viewed as an extension of the Levenberg–Morrison–Marquardt algorithm for nonlinear least squares. Global convergence of trust-region algorithms for nonlinear equations is studied and local convergence analyses are given.
- Y. Yuan (1999). A review of trust region algorithms for optimization. Technical Report ICM-99-038, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, Beijing, China.
- Summary.** A survey of trust-region methods is presented, with special emphasis on the possible subproblem solution techniques. Convergence properties for the unconstrained case are reviewed and techniques such as backtracking and nonmonotone descent and second-order correction are briefly discussed.
- Y. Yuan (2000). On the truncated conjugate-gradient method. *Mathematical Programming*, **87**(3), 561–573.
- Summary.** It is shown that the model reduction obtained by applying the Steihaug–Toint algorithm on a convex quadratic model is at least half of that obtained by the exact minimizer of the model within the trust region.
- E. H. Zarantonello (1971). Projections on convex sets in Hilbert space and spectral theory. In E. H. Zarantonello, ed., *Contributions to Nonlinear Functional Analysis*, pp. 237–424, Academic Press, London.
- J. Zhang (1989). Superlinear convergence of a trust region-type successive linear-programming method. *Journal of Optimization Theory and Applications*, **61**(2), 295–310.

**Summary.** The convergence rate of the successive linear programming method suggested by Zhang, Kim, and Lasdon (1985) is discussed for composite nondifferentiable optimization problems. A superlinear rate is assured under a growth condition, and it is further strengthened to a quadratic rate if the inside function is twice differentiable. Several sufficient conditions are given that make the growth condition true. The conditions can be relaxed considerably in practical use.

J. Zhang, N. H. Kim, and L. S. Lasdon (1985). An improved successive linear programming algorithm. *Management Science*, **31**, 1312–1331.

J. Zhang and C. Xu (1999a). A class of indefinite dogleg path methods for unconstrained optimization. *SIAM Journal on Optimization*, **9**(3), 646–676.

**Summary.** A variant of the dogleg method for approximately solving the trust-region subproblem is proposed. This variant is adequate for the case where the subproblem is nonconvex. In this case, it defines a family of paths that use a direction of negative curvature obtained from the Bunch–Parlett factorization of the Hessian. Global convergence to unconstrained second-order stationary points is proved for the resulting trust-region method, as is quadratic convergence of the associated version of Newton's method. Numerical results are shown.

J. Zhang and C. Xu (1999b). A projected indefinite dogleg-path method for equality constrained optimization. *BIT*, **39**(3), 555–578.

**Summary.** A two-step trust-region algorithm is presented for the solution of optimization problems with nonlinear equality constraints. The method is based on that of Fontecilla (1990) but uses an indefinite dogleg strategy in the null-space of the constraints Jacobian as a way to obtain an approximate solution of the quadratic programming subproblem. Global and locally two-step superlinear convergence are proved and some numerical experiments shown.

J. Zhang and C. Xu (1999c). Trust region dogleg path algorithms for unconstrained minimization. *Annals of Operations Research*, **87**, 407–418.

**Summary.** Trust-region algorithms using curvilinear search to approximately solve the possibly nonconvex subproblem are proved to be globally convergent to first- and second-order critical points. The rate of convergence is quadratic under typical assumptions.

J. Zhang, C. Xu, and L. Du (1998). A more efficient variation of an indefinite dogleg path method. In *Operations Research and Its Applications. Third International Symposium, ISORA '98*, pp. 428–434, World Publishing, Beijing, China.

**Summary.** A scaled version of the author's previous indefinite dogleg path method is considered. The main advantage is that the scaled subproblem has a 1 by 1 or 2 by 2 block diagonal Hessian, so that the solution of the Newton equations and directions of negative curvature are simple to obtain.

J. Zhang and D. Zhu (1990). Projected quasi-Newton algorithm with trust-region for constrained optimization. *Journal of Optimization Theory and Applications*, **67**, 369–393.

**Summary.** A trust-region-type, two-sided, projected quasi-Newton method is proposed, which preserves the local two-step superlinear convergence of the original algorithm of Nocedal and Overton (1985) and also ensures global convergence. The proposed subproblem is as simple as the one used when solving unconstrained problems by trust-region strategies.

J. Zhang and D. Zhu (1994). A projective quasi-Newton method for nonlinear optimization. *Journal of Computational and Applied Mathematics*, **53**(3), 291–307.

- Summary.** A trust-region method for nonlinear optimization problems with equality constraints is proposed that incorporates quadratic subproblems in which orthogonal projective matrices of the Jacobian of constraint functions are used to replace QR decompositions. As QR decomposition does not ensure continuity, but the projective matrix does, the convergence behaviour of the new method is studied by exploiting the continuity of these matrices. A one-step superlinear convergence rate is established.
- J. Zhang and D. Zhu (1999). A nonmonotonic trust region method for constrained optimization problems. *Journal of the Australian Mathematical Society (Series B)*, **40**(4), 542–567.
- Summary.** A nonmonotonic, reduced Hessian trust-region method is used to solve equality constrained nonlinear optimization problems. Approximate solutions to the trust-region subproblems are allowed. Although theory dictates that second-order correction steps should be used to overcome the Maratos effect, a suitable scheme is developed to ensure that they are only used when absolutely necessary. Global convergence at a local superlinear rate is established, and the resulting algorithm performs well in practice.
- J. Zhang, D. Zhu, and Y. Fan (1993). A practical trust region method for equality constrained optimization problems. *Optimization Methods and Software*, **2**(1), 45–68.
- Summary.** An easy-to-implement algorithm for solving general nonlinear optimization problems with nonlinear equality constraints is proposed that uses a reduced Hessian matrix. The quadratic trust-region subproblems are solved approximately. The calculation of correction steps, which are necessary from the theoretical point of view to overcome the Maratos effect but prove costly in practice, is avoided in most cases by a suitable test. Global and superlinear convergence are proved. Numerical results are reported.
- Y. Zhang (1992). Computing a Celis-Dennis-Tapia trust-region step for equality constrained optimization. *Mathematical Programming*, **55**(1), 109–124.
- Summary.** An approach to minimizing a convex quadratic function subject to two quadratic constraints is studied. This problem stems from computing a trust-region step for a sequential quadratic programming algorithm proposed by Celis, Dennis, and Tapia (1985) for equality constraint optimization. The approach taken is to reformulate the problem as a univariate nonlinear equation  $\phi(\mu) = 0$ , where the function  $\phi(\mu)$  is continuous, at least piecewise differentiable, and monotone. Well-established methods then can be readily applied. An extension of this approach to a class of nonconvex quadratic functions is considered, and it is shown that the approach is applicable to reduced Hessian sequential quadratic programming algorithms. Numerical results are presented.
- Y. Zhang (1994). On the convergence of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM Journal on Optimization*, **4**(1), 208–227.
- M. Zhao and X. Wang (1993). Model trust region technique in parallel Newton method for training neural networks. In *IEEE International Symposium on Circuits and Systems (ISCAS 93)*, Vol. 4, pp. 2399–2402, IEEE, New York.
- Summary.** The double-dogleg trust-region approach of unconstrained minimization is introduced into the parallel Newton's (PN) algorithm, which uses a recursive procedure for computing both the Hessian matrix and the Newton direction. The input weights of each neuron in the network are updated after each presentation of the training data with a global strategy. Experimental results indicate that the double-dogleg trust-region approach is superior to the linesearch technique in the PN algorithm and that the PN algorithm with both global strategies exhibits better convergence performance than back propagation.

- F. Zhou and Y. Xiao. A class of nonmonotone stabilization trust region methods. *Computing*, **53**(2), 119–136, 1994.

**Summary.** A class of trust-region methods for unconstrained optimization is presented that uses a nonmonotone stabilization strategy. Under some regularity conditions, the convergence properties of these methods are discussed. Extensive numerical results are reported.

- G. Zhou and J. Si (1998). Advanced neural-network training algorithm with reduced complexity based on Jacobian deficiency. *IEEE Transactions on Neural Networks*, **9**(3), 448–453.

**Summary.** A supervised training method for neural networks based on Jacobian rank deficiency is formulated in the spirit of the Gauss–Newton algorithm. The new method aims at improving convergence properties compared to the Levenberg–Morrison–Marquardt method, while reducing the memory and computation complexities in supervised training of neural networks. Extensive simulation results demonstrate the superior performance of the new algorithm over the Levenberg–Morrison–Marquardt algorithm.

- G. Zhou and J. Si (1999). Subset based training and pruning of sigmoid neural networks. *Neural Networks*, **12**(1), 79–89.

**Summary.** Two trust-region algorithms, subset-based training (SBT) and subset-based training and pruning (SBTP), are developed using the fact that the Jacobian matrices in sigmoid network training problems are usually rank deficient. The weight vectors are divided into two parts during training, according to the Jacobian rank sizes. These two algorithms have convergence properties similar to those of the Levenberg–Morrison–Marquardt method but with fewer memory requirements. Furthermore, the SBTP combines training and pruning of a network into one comprehensive procedure.

- J. Zhou and A. L. Tits (1993). Nonmonotone line search for minimax problems. *Journal of Optimization Theory and Applications*, **76**(3), 455–476.

- C. Zhu (1996). Asymptotic convergence analysis of some inexact proximal point algorithms for minimization. *SIAM Journal on Optimization*, **6**(3), 626–637.

- C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal (1997). Algorithm 78: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, **23**(4), 550–560.

- D. Zhu (1992). Convergence of a projected gradient method with trust region for nonlinear constrained optimization. *Optimization*, **23**(3), 215–235.

**Summary.** A globally convergent trust-region projected-gradient algorithm is described that uses a nondifferentiable merit function.

- D. Zhu (1995). A nonmonotonic trust region technique for nonlinear constrained optimization. *Journal of Computational Mathematics*, **13**(1), 20–31.

**Summary.** A nonmonotonic method for problems with equality constraints is proposed by introducing a nonsmooth merit function and a correction step. It is proved that all accumulation points of the iterates generated are Kuhn–Tucker points and that the algorithm is Q-superlinearly convergent.

- D. Zhu (1999). A family of generalized projected gradient methods with mixing strategy for convex constrained optimization. Technical Report (na), Department of Mathematics, Shanghai Normal University, Shanghai, China.

- Summary.** A globally convergent trust-region method is proposed for problems with convex constraints; it combines the strategies of Conn et al. (1993) with the nonmonotone technique of Deng, Xiao, and Zhou (1993).
- M. Zhu, J. L. Nazareth, and H. Wolkowicz (1999). The quasi-Cauchy relation and diagonal updating. *SIAM Journal on Optimization*, **9**(4), 1191–1204.
- Summary.** The quasi-Cauchy (QC) relation, a weak quasi-Newton relation in which an added restriction that full matrices are replaced by diagonal matrices, is justified and explored. Two basic variational techniques for updating diagonal matrices that satisfy QC are formulated. In practice, QC is shown to significantly accelerate the convergence of gradient methods. A connection between QC-diagonal updating and trust-region techniques is highlighted.
- T. Zhu and L. D. Brown (1987). Two-dimensional velocity inversion and synthetic seismogram computation. *Geophysics*, **52**(1), 37–50.
- Summary.** A trust-region method is used to estimate velocity and interface geometries of two-dimensional media from deep-reflection data, where the velocity structure is represented by finite elements. The traveltimes and derivative matrix required to solve the least-squares problem are computed by ray tracing.
- M. Zupke (1997). Trust-Region-Verfahren zur Lösung nichtlinearer Komplementaritätsprobleme. Master's thesis, Institute of Applied Mathematics, University of Hamburg, Germany.

**Summary.** The nonlinear complementarity problem is reformulated as a nonsmooth system of equations by using a recently introduced nonlinear complementarity problem function. A trust-region-type method that allows an inexact solution of the trust-region subproblem is then applied to the resulting system of equations. It is shown that the algorithm is well defined for a general nonlinear complementarity problem and that it has some good global and local convergence properties. Numerical results show the advantage of using the nonmonotone technique proposed by Toint (1997).

# Subject and Notation Index

**Note:** A boldface number indicates the page on which the subject is introduced. An underlined number indicates that the subject is a primary topic in the section or subsection starting on the page given.

$\cdot^+$ , <b>15</b>	$\alpha_\eta$ , <b>596</b>
$\cdot^-$ , <b>15</b>	$\alpha_\omega$ , <b>598</b>
$\cdot^{\frac{1}{2}}$ , <b>17</b>	$\alpha_{t,k}$ , <b>338</b>
$\cdot^T$ , <b>17</b>	$\beta_\eta$ , <b>598</b>
$\sqrt{\cdot}$ , <b>17</b>	$\beta_\omega$ , <b>598</b>
$\cdot_S$ , <b>15</b>	$\beta_{\{k\}}$ , <b>262</b>
$[\cdot]_S$ , <b>15</b>	$\beta_k^T$ , <b>666</b>
$[\cdot]_i$ , <b>15</b>	$\beta_k$ , <b>124</b>
$[\cdot]_{i,j}$ , <b>15</b>	$\beta_k^\pi$ , <b>251</b>
$[\cdot]$ , <b>16</b>	$\gamma_1$ , <b>116</b>
$\lceil \cdot \rceil$ , <b>16</b>	$\gamma_2$ , <b>116</b>
$\cdot^\perp$ , <b>360</b>	$\gamma_3$ , <b>159</b>
$\cdot_{i,j}$ , <b>15</b>	$\gamma_4$ , <b>159</b>
$\langle \cdot, \cdot \rangle$ , <b>21</b>	$\gamma_5$ , <b>284</b>
$\ \cdot\ $ , <b>16</b>	$\delta_{i,j}$ , <b>15</b>
$\ \cdot\ _1$ , <b>21</b>	$\epsilon_M$ , <b>299</b>
$\ \cdot\ _2$ , <b>21</b>	$\eta$ , <b>598</b>
$\ \cdot\ _p$ , <b>21</b>	$\eta_1$ , <b>116</b>
$\ \cdot\ _\infty$ , <b>21</b>	$\eta_2$ , <b>116</b>
$\ \cdot\ _H$ , <b>22</b>	$\eta^{CS}$ , <b>680</b>
$\ \cdot\ _k$ , <b>116</b>	$\theta(x)$ , <b>700</b>
$\ \cdot\ _{[k]}$ , <b>259</b>	$\theta_k$ , <b>712</b>
$\ \cdot\ _{\{k\}}$ , <b>260</b>	$\vartheta(x)$ , <b>722</b>
$ \cdot $ , <b>17</b>	$\vartheta_k$ , <b>722</b>
$\Delta_{i,k}$ , <b>361</b>	$\kappa_\alpha$ , <b>337</b>
$\Delta_k$ , <b>115</b>	$\kappa_\eta$ , <b>596</b>
$\Delta_k^T$ , <b>660</b>	$\kappa_{aat}$ , <b>618</b>
$\Delta_{\max}$ , <b>159</b>	$\kappa_{aeg}$ , <b>308</b>
$\Delta_k^{\min}$ , <b>361</b>	$\kappa_{aeh}$ , <b>319</b>
$\Theta(\cdot)$ , <b>50</b>	$\kappa_{aha}$ , <b>618</b>
$\Pi(x)$ , <b>330</b>	$\kappa_{amm}$ , <b>132</b>
$\Phi(x, y, \mu)$ , <b>594</b>	$\kappa_{bbh}$ , <b>494</b>
$\Phi_y(x, \mu)$ , <b>575</b>	$\kappa_{bbmh}$ , <b>499</b>
$\nabla^2$ , <b>27</b>	$\kappa_{bck}$ , <b>128</b>
$\nabla_x$ , <b>27</b>	$\kappa_{bdm}$ , <b>675</b>
$\nabla_{xx}$ , <b>27</b>	$\kappa_{bik}$ , <b>581</b>
$\nabla_{xxx}$ , <b>319</b>	$\kappa_{bnc}$ , <b>703</b>
$\alpha$ , <b>600</b>	$\kappa_{bns}$ , <b>666</b>
	$\kappa_{boa}$ , <b>606</b>
	$\kappa_{bon}$ , <b>674</b>
	$\kappa_{bos}$ , <b>689</b>
	$\kappa_{brs}$ , <b>702</b>
	$\kappa_{bsc}$ , <b>662</b>

$\kappa_{\text{btm}}$	676	$\kappa_{\text{umt}}$	319
$\kappa_{\text{cdn}}$	273	$\kappa_{\text{zul}}$	525
$\kappa_{\text{cmp}}$	774	$\kappa_{\text{zuu}}$	525
$\kappa_{\text{end}}$	319	$\lambda_i[H]$	18
$\kappa_{\text{dcp}}$	128	$\lambda^M$	172
$\kappa_{\text{dns}}$	697	$\lambda_i[H, M]$	19
$\kappa_{\text{drs}}$	697	$\lambda_\ell(x)$	326
$\kappa_{\text{easy}}$	194	$\lambda_{\max}[H]$	18
$\kappa_{\text{egg}}$	280	$\lambda_{\min}[H]$	18
$\kappa_{\text{epp}}$	453	$\mu$	492
$\kappa_{\text{fgr}}$	206	$\mu(\cdot)$	277
$\kappa_{\text{frd}}$	453	$\nu_k^C$	126
$\kappa_{\text{hard}}$	196	$\nu_k^E$	148
$\kappa_{\text{lblb}}$	503	$\nu_k^{\text{MC}}$	268
$\kappa_{\text{lbd}}$	135	$\nu_k^W$	252
$\kappa_{\text{lbf}}$	121	$\xi^N$	695
$\kappa_{\text{lbfg}}$	135	$\xi^T$	664
$\kappa_{\text{lbes}}$	453	$\pi(k, x_k)$	250
$\kappa_{\text{lca}}$	669	$\pi(x)$	330
$\kappa_{\text{lcc}}$	482	$\pi_k$	251
$\kappa_{\text{lcg}}$	668	$\rho^C(x, p, s)$	416
$\kappa_{\text{lch}}$	151	$\rho_k$	116
$\kappa_{\text{lsd}}$	256	$\varrho_i(t)$	210
$\kappa_{\text{mdb}}$	500	$\varrho_i(t, \lambda)$	212
$\kappa_{\text{mdc}}$	131	$\sigma_i[A]$	18
$\kappa_{\text{mma}}$	338	$\sigma_k$	352
$\kappa_{\text{mnms}}$	390	$\sigma_{\max}[A]$	18
$\kappa_{\text{mqd}}$	144	$\sigma_{\max}$	675
$\kappa_{\text{msd}}$	560	$\sigma_{\min}[A]$	18
$\kappa_{\text{nfp}}$	332	$\sigma_{t,k}$	354
$\kappa_{\text{nmd}}$	149	$\tau_k$	148
$\kappa_{\text{opt}}$	194	$\phi(\lambda)$	182
$\kappa_{\text{smd}}$	364	$\phi(x, \mu)$	492
$\kappa_{\text{smh}}$	141	$\phi(x, \sigma)$	610
$\kappa_{\text{snc}}$	148	$\phi^{\log}(x, \mu)$	491
$\kappa_{\text{sod}}$	151	$\varphi_k$	283
$\kappa_{\text{sos}}$	618	$\psi(x, \sigma)$	671
$\kappa_t$	338	$\omega$	598
$\kappa_{\text{tmd}}$	666	$\omega_k(h, x, w)$	271
$\kappa_{\text{tr}}$	194	$\mathcal{A}$	729
$\kappa_{\text{ubc}}$	150	$\mathcal{A}(\mathcal{L})$	463
$\kappa_{\text{ubh}}$	133	$\mathcal{A}(x)$	39
$\kappa_{\text{ubm}}$	483	$\mathcal{A}_*$	468
$\kappa_{\text{ubs}}$	128	$\mathcal{B}_\epsilon(x)$	25
$\kappa_{\text{ubt}}$	733	$\mathcal{B}_k$	115
$\kappa_{\text{uch}}$	482	$\mathcal{B}_k^{(i)}$	361
$\kappa_{\text{ucj}}$	483	$\mathcal{B}_k^{\min}$	363
$\kappa_{\text{uem}}$	371	$\mathcal{C}$	441
$\kappa_{\text{ufg}}$	333	$\mathcal{C}_{\mathcal{E}}$	542
$\kappa_{\text{ufh}}$	121	$C^k$	27
$\kappa_{\text{ufm}}$	668	$\mathcal{E}_1$	180
$\kappa_{\text{uhn}}$	688	$\mathcal{F}$	726
$\kappa_{\text{umh}}$	122	$\mathcal{F}(x)$	723

$\mathcal{F}_*$ , <b>473</b>	$\partial f(x)$ , <b>33</b>
$\overline{G}_k$ , <b>556</b>	$\text{dist}(y, \mathcal{Z})$ , <b>493</b>
$H(\lambda)$ , <b>172</b>	$e_i$ , <b>16</b>
$H(x, y)$ , <b>624</b>	$f'_d(x)$ , <b>33</b>
$H^{\text{CS}}$ , <b>649</b>	$f(x)$ , <b>115</b>
$H^N(x)$ , <b>665</b>	$f_d^o(x)$ , <b>33</b>
$H_k$ , <b>117</b>	$g^{\text{CS}}$ , <b>649</b>
$H_k^N$ , <b>666</b>	$g^N(x)$ , <b>665</b>
$I$ , <b>16</b>	$g_{i,k}$ , <b>361</b>
$I_n$ , <b>16</b>	$g_k$ , <b>117</b>
$\mathcal{K}(H, r_0, j)$ , <b>81</b>	$j_c$ , <b>128</b>
$\mathcal{K}^0$ , <b>35</b>	$\ell(x, y)$ , <b>41</b>
$\mathcal{K}_\epsilon$ , <b>253</b>	$m(x, H, \sigma, s)$ , <b>660</b>
$K_k^{\text{CS}}$ , <b>682</b>	$m(x, H, s)$ , <b>638</b>
$\mathcal{L}_*$ , <b>460</b>	$m(x, p, s)$ , <b>408</b>
$L^1(\Omega)$ , <b>277</b>	$m(x, s, \mu)$ , <b>637</b>
$L^p(\Omega)$ , <b>277</b>	$m^N(x, n)$ , <b>660</b>
$L^\infty(\Omega)$ , <b>277</b>	$m^T(x, H, t)$ , <b>660</b>
$M^{\text{TYP}}$ , <b>795</b>	$\max[\cdot, \cdot]$ , <b>15</b>
$M_k$ , <b>366</b>	$\text{mid}(\cdot, \cdot, \cdot)$ , <b>15</b>
$N(\cdot)$ , <b>477</b>	$m_{i,k}(x)$ , <b>361</b>
$\mathcal{N}(x)$ , <b>35</b>	$\min[\cdot, \cdot]$ , <b>15</b>
$\mathcal{N}_i$ , <b>360</b>	$m_k(x)$ , <b>115</b>
$N_i^{[\ell]}(x)$ , <b>324</b>	$m_k^f(x)$ , <b>556</b>
$N_k$ , <b>366</b>	$m_{k,j}^b(x)$ , <b>498</b>
$O(\cdot)$ , <b>50</b>	$m_{k,j}^f(x)$ , <b>499</b>
$\mathcal{O}_+$ , <b>491</b>	$m_k^f(x)$ , <b>475</b>
$\mathcal{O}_\epsilon(x)$ , <b>25</b>	$n^C(x)$ , <b>658</b>
$P_{\mathcal{R}_i}(\cdot)$ , <b>360</b>	$n_k$ , <b>696</b>
$P_{\mathcal{X}}(x)$ , <b>36</b>	$o(\cdot)$ , <b>50</b>
$\mathcal{Q}_k(\delta)$ , <b>308</b>	$p(t, x)$ , <b>444</b>
$\mathcal{R}$ , <b>730</b>	$q(x, H, s)$ , <b>660</b>
$\mathcal{R}_i$ , <b>360</b>	$\text{ri}\{\mathcal{S}\}$ , <b>26</b>
$S$ , <b>117</b>	$s(t)$ , <b>210</b>
$\mathcal{T}(x)$ , <b>35</b>	$s(t, \lambda)$ , <b>212</b>
$\mathcal{V}$ , <b>117</b>	$\text{sgn}(\cdot)$ , <b>16</b>
$\mathcal{V}(x, \delta)$ , <b>464</b>	$s^{\text{CS}}$ , <b>643</b>
$W^{k,p}(\Omega)$ , <b>279</b>	$s^M$ , <b>169</b>
$Y^{[\ell]}$ , <b>324</b>	$s^{\text{ST}}$ , <b>208</b>
$\varphi$ -divergence, <b>120</b>	$s_k^S$ , <b>590</b>
$\text{aff}\{\mathcal{S}\}$ , <b>26</b>	$s_k$ , <b>116</b>
$b(x, \mu)$ , <b>492</b>	$s_k^{\text{CS}}$ , <b>643</b>
$b^{R(\alpha)}(x, \mu)$ , <b>517</b>	$s_k^{\text{FC}}$ , <b>681</b>
$b^{\log}(x, \mu)$ , <b>492</b>	$s_k^{\text{MA}}$ , <b>387</b>
$c_{\mathcal{I}}^+$ , <b>427</b>	$s_k^{\text{PC}}$ , <b>681</b>
$c_{\mathcal{I}}^-$ , <b>427</b>	$s_k^{\min}$ , <b>363</b>
$c^{\mathcal{I}-}$ , <b>427</b>	$s^y$ , <b>644</b>
$c^{\mathcal{I}+}$ , <b>427</b>	$t_k$ , <b>665</b>
$\text{cl}\{\mathcal{S}\}$ , <b>26</b>	$t_k^{\text{AE}}$ , <b>150</b>
$\text{co}\{\mathcal{S}\}$ , <b>30</b>	$t_k^C$ , <b>125</b>
$\partial(Y)$ , <b>323</b>	$t_k^{\text{CC}}$ , <b>506</b>
$\partial\mathcal{C}$ , <b>492</b>	$t_k^E$ , <b>148</b>
$\det(\cdot)$ , <b>18</b>	$t_k^{\text{GE}}$ , <b>482</b>

- $t_k^{\text{MC}}$ , **269**  
 $u_i[H]$ , **18**  
 $u_i[H, M]$ , **19**  
 $v_k$ , **280**  
 $x^{\text{TYP}}$ , **794**  
 $x_k^S$ , **589**  
 $x_k(t)$ , **128**  
 $x_k^{\text{AC}}$ , **128**  
 $x_k^{\text{AE}}$ , **150**  
 $x_k^C$ , **125**  
 $x_k^C(t)$ , **124**  
 $x_k^{\text{CC}}$ , **506**  
 $x_k^{\text{CE}}$ , **510**  
 $x_k^{\text{CSC}}$ , **559**  
 $x_k^{\text{CSE}}$ , **563**  
 $x_k^E$ , **148**  
 $x_k^M$ , **131**  
 $y_k^{\text{CS}}$ , **644**  
 $y^{\text{LS}}(x)$ , **474**  
 $y^{\text{TYP}}$ , **798**  
 $y_k^E$ , **602**  
 $y_k^S$ , **589**  
 $y_k^F$ , **579**  
 $y_y^F(x_k, \mu_k)$ , **588**  
 $y_k^{\text{LS}}$ , **475**
- acceptable  
approximation, **194**, 217  
for the filter, **726**
- accuracy  
dynamic, **399**, **400**  
algorithm, **400**  
of finite differences, 300  
of gradient, 296, 299  
of objective function, **399**, **405**  
of subproblem solution, 554
- active  
consistently, *see* subsequence, consistently active  
constraint, *see* constraint, active; constraint, asymptotically active  
function, **431**  
set, 460; *see also* set, active  
method, *see* working set method
- affine  
hull, **26**  
scaling, **550**  
algorithm, **550**  
set, **26**  
step, **567**  
subspace, 542
- algorithm, **1**  
affine-scaling, **550**
- augmented Lagrangian, **595**  
alternative, **602**  
tolerances, **598**  
backtracking, **381**  
barrier  
general, **494**  
primal (inner, first), **499**  
primal (inner, second), **504**  
primal (outer), **511**  
primal-dual (general constraints, inner), **538**  
primal-dual (general constraints, outer), **539**  
primal-dual (inner), **520**  
primal-dual (linear constraints, inner), **543**  
primal-dual (outer), **521**  
reciprocal, **517**  
basic trust region, **115**, **116**, **133**, **139**  
BTR, *see* algorithm, basic trust region  
Byrd–Omojokun, **698**  
Coleman–Li, **554**, **558**  
modified  $\rho_k$ , **565**  
conjugate directions, generation of, **81**  
conjugate gradient, **82**  
for least-squares problems, **107**  
preconditioned, **88**  
truncated, *see* conjugate gradient method, truncated  
for convex constraints, **452**  
generalized finite differences, **330**  
generalized Lanczos trust-region, **228**, 399, 666, 697  
Lanczos, **95**  
preconditioned, **104**  
linesearch  
double, **376**, **378**  
generalized Cauchy point, **455**  
Newton fundamental polynomials, **325**, 326–336  
Newton’s method, **51**, **52**, **122**, **147**, **590**  
for constrained problems, 624  
to solve secular equation, **185**  
nonmonotone, **347**, **349**, **474**  
modified  $\rho_k$ , **357**  
nonsmooth problems, **413**  
with correction step, **425**  
nonsmooth, with correction step, **769**  
penalty function, quadratic, **583**  
SQP  
Byrd–Omojokun with second-order convergence, **709**  
filter, **727**

- infeasibility reducing, **714**
- relaxed linearization, **666**
  - relaxed linearization with second-order convergence, **692**
  - relaxed linearization with second-order correction, **680, 706**
  - relaxed linearization with a specific second-order correction, **686**
  - single-step filter, **743**
- Steihaug–Toint, **202, 205**, 384, 386, 399, 666, 697
  - quality of, **208**
- structured, **359, 368**
  - radii update, **366**
- to choose  $\Delta_0$ , **787**
- to find model minimizer, **193**
  - enhancements, **197**
  - termination rules, **194, 196**
  - using eigensolution, **199**
- to find the generalized Cauchy point, **791**
- TRAM, **568**
- updating
  - the interval of uncertainty, **188**
  - the safeguarding parameters, **189**
- with conditional models, **307, 310**
  - modified criticality test, **321**
- with dynamic accuracy, **400, 402**
- with internal doubling, **394, 394**
- with magical steps, **387, 388**
- analytic centre, **549**
- assumptions, **121, 803**
  - numbering scheme, xvii
- asymptotically
  - active constraint, **513, 514–517**
  - bounded, **50**
  - similar, **50**
  - smaller, **50**
- augmented
  - Lagrangian, **575, 617, 637**
    - as a merit function, **711**
    - method, **593**
  - systems, **57, 69–71, 110, 634**
- automatic differentiation, **301**
- back
  - solution, **58**
  - substitution, **58**
- backtracking, **128, 151, 380, 482, 627**
  - algorithm, **381**
- ball, closed or open, **25**
- Banach space, **274, 277**
- barrier
  - function, **492**
- logarithmic, **491, 491, 511, 518, 527, 536**
  - reciprocal, **517**
- Lagrangian, **497**
- modified, **496**
- parameter, **491, 492, 498, 511, 518**
- self-concordant, **497, 498**
- shift, **496**
- shifted, **496**
- term, **492, 492**
  - logarithmic, **491, 492, 511, 518, 527, 536**
  - reciprocal, **517**
  - weight, **496**
- BCCSS effect, *see* Byrd–Coleman–Conn–Schnabel–Shultz effect
- BECTR, **720**
- BFGS update, *see* Broyden–Fletcher–Goldfarb–Shanno update
- bidual norm, **260, 527**
- binding constraint, *see* constraint, active
- BOX-QUACAN, **459**
- Bregman distance, **120**
- Broyden–Fletcher–Goldfarb–Shanno (BFGS) update, **281, 296**
- BTR algorithm, *see* algorithm, basic trust region
- bundle method, **422**
- Byrd–Coleman–Conn–Schnabel–Shultz (BCCSS) effect, **691**
- Byrd–Omojokun-like approaches, **694**
- canonical basis vector, **16**
- Cauchy
  - arc, **123, 124**
    - modified, **266**
  - interlacing property, **19**
  - point, **123, 125, 139, 169, 262, 384, 759**
    - approximate, **128, 128**
    - constrained, **506, 559**
    - constrained scaled, **559**
    - for normal step subproblem, **696**
    - for tangential step subproblem, **665**
    - generalized, **453, 453, 724, 789**
      - approximate, **453**
      - computation of, **789**
    - in infinite dimensions, **277**
    - modified, **265, 266, 376**
    - scaled, **271, 559**
      - step, **169, 207, 411**
  - Cauchy–Schwarz inequality, **21**
  - Celis–Dennis–Tapia-like approaches, **711**
  - central differences, **298**
  - Chebyshev polynomial, **85**

- Cholesky factorization, 62, **63**  
 Clarke's theorem, **33**, 411, 773  
 Coleman–Li algorithm, **554**  
 complementarity  
     function, **771**  
     merit function, **772**, 776  
     problem, **770**, 770  
     generalized, **770**  
     linear, **771**  
     mixed, **777**  
     nonlinear, **770**  
         monotone, **776**  
 complementary slackness, **41**  
     strict, **42**  
 composite step, **387**; *see also* step, composite  
 concave function, **31**  
 condition number, **61**, 82, 85, 86, 89  
 conditioning  
     ill-, **53**, **61**  
     of augmented system  
         for quadratic penalty methods, 587  
     of barrier function, 495–497  
     of Hessian, 86  
         of augmented Lagrangian function, 593  
         of quadratic penalty function, 586  
     well-, **61**  
 cone, **34**  
     normal, **35**, 46, 444, 449, 460  
     polar, **35**, 449  
     tangent, **35**, 449–451, 467, 489  
 conjugate  
     basis, **77**  
     directions, 77  
         generation of, **78**, **81**  
     vectors, **77**  
 conjugate gradient method, **78**, 82  
     algorithm, **82**  
         for least-squares problems, **107**  
         for transformed problem, **87**  
         preconditioned, **88**  
             in null-space, **109**  
             projected, **109**  
     convergence of, **83**  
          $< n$ -step, **84**  
         preconditioned, **89**  
     n-step, **83**  
     error bounds, **85**  
         preconditioned, **89**  
     preconditioned, **89**  
     for problems with constraints, **108**  
     preconditioned, **86**, **88**  
         projected, **109**, **207**  
     to find approximate model minimizer, 202  
     truncated, **202**, **205**, 384, 386, 399, 666,  
         697  
 conjugate gradient path, **204**  
     parametrization, 210  
     shifted, **212**  
 constrained problem, **37**, 39  
 constraint  
     active, **39**, 460–474  
     asymptotically active, **513**, 514–517  
     convex, 491, **498**, **504**, **511**, **517**, **518**, **527**  
     equality, **39**  
     equilibrium, 422, 777  
     inactive, **39**, 755  
     inequality, **39**, 536  
     linear, 479, 516, **542**  
     qualification, **39**  
 control  
     optimal, 10, 274, 279, 498  
     distributed, 279  
 convergence, **50**  
     global, **133**  
     rate, **50**, 169, 186, 194, 199, 206, 207,  
         247, 626, 630, 632, 633, 641  
     of nonmonotone algorithm, 354  
     of algorithm with linear constraints,  
         474  
     of augmented Lagrangian method, 607  
     of basic algorithm, 147  
     of methods for complementarity prob-  
         lems, 779  
     of primal-dual algorithm, 550  
     of quadratic penalty method, 593  
     of SQP method, 625, 632  
     of the Coleman–Li algorithm, 569  
     of trust-region SQP method, 687, 707  
 Q-linear, **50**  
 Q-quadratic, **51**  
 Q-superlinear, **51**  
 R-linear, **51**  
 R-p, **51**  
 two-step Q-superlinear, **593**  
 to first-order critical points, *see* critical  
     point  
 to second-order critical points, *see* criti-  
     cal points  
 convex  
     combination, **30**  
     cone, **34**  
     function, **31**  
         strictly, **31**  
     hull, **30**  
     polyhedron, 479  
     problem, 44

- program, **31**, 44
- set, 30, 450
- correction step, **392**
- CPLEX, 636
- critical point
  - constrained
    - first-order, **41**, 551, 552, 555, 557, 561, 568, 771
    - convergence to, 459, 510, 563, 569, 583, 598, 638, 678, 705
    - second-order, 512, 539
    - convergence to, 488, 510, 566, 569
    - strong second-order, **42**, 515
    - convergence to, 692, 709
    - weak second-order, **43**, 515, 540, 557
  - nonsmooth first-order, **48**
  - unconstrained
    - first-order, **38**, 503, 522, 524
    - convergence to, 133, 137, 254, 274, 292, 317, 353, 375, 393, 404, 753
    - second-order, **38**, 522, 565
    - convergence to, 153, 155, 157, 159, 258, 321, 354, 375, 404, 753
- criticality measure, *see* measure, criticality
- curvature, **32**
  - negative, **32**, 148, 150, 481
  - direction of, **32**
  - positive, **32**
  - direction of, **32**
- CUTE, 343, 750, 781, 783
- cycling, 634, 635
- damping, 8
- derivative
  - approximate, 280
  - directional
    - generalized, **33**
    - one-sided, **33**
  - Fréchet, 274
  - free optimization, 322
  - gradient, *see* gradient
  - Hessian, *see* Hessian matrix
  - Jacobian matrix, **27**
- descent method, **347**
- determinant, **18**
- differences, finite, 297
  - central, **298**
  - forward, **297**
  - generalized, **326**, 328–331
  - stepsizes, **297**
- Dikin ellipsoid, 550
- direction
  - of negative curvature, 32
  - of positive curvature, 32
- distributed control, 279
- dogleg method, **219**
  - double, **219**, 758
  - dominated  $(\vartheta, f)$ -pair, **726**
- dual
  - pairing, **275**
  - space, **274**
  - variable, *see* variable, dual
  - vector norm, **21**
- dynamic accuracy, 399
- easy case, **181**, 192, 194
- eigendecomposition, **17**
- eigenpair, **16**
  - generalized, **19**
- eigenpoint, 147, **148**, 207
  - approximate, **150**
  - computing, 230
  - constrained, **510**
  - generalized, 481, **482**
  - scaled, **265**
  - constrained, **563**
- eigenproblem, **16**
  - generalized, **19**
- eigenproblems, 16
- eigenvalue, **16**
  - generalized, **19**
- eigenvector, **16**
  - generalized, **19**
- EISPACK, 20
- element, **360**
  - function, **360**
  - meaningful, **366**
  - negligible, **366**
- elemental variable, **360**
- ellipsoid, Dikin, 550
- equality-constrained quadratic programming (EQP), **631**
- equations
  - linear, 55
  - nonlinear, 271, 749, 759
  - ordinary differential, 162, 280
  - partial differential, 10, 90, 279, 280
  - secant, **281**
- error
  - discretization, 610
  - in gradient, **280**
    - relative, **280**
  - in objective function value, 399
  - in solution, **83**
  - rounding, **298**
  - typographical, xv, 1–959
- ETR, 710
- evolution triple, **279**

- face, **39**, 461  
     optimal, **473**, 474
- factorization  
     Cholesky, **62**, **63**  
     cost of, 197  
     following low-rank update, 198  
     for solving the secular equation, 184
- complete orthonormal, **69**, 72  
 $LDL^T$ , **63**  
 $LQ$ , **59**, **69**  
     modified, 67  
     absolute-value, **240**  
     Cholesky, **67**  
     symmetric indefinite, **65**, 65
- feasible  
     region, 37, 441  
     set, **37**  
     solution to the equality problem (FSEP), **246**
- filter, **726**  
 filterSQP, 744
- finite differences, 297  
     central, **298**  
     forward, **297**  
     generalized, **326**, 328–331
- first-order Lagrange multiplier estimate, *see* Lagrange
- Fischer–Burmeister function, **771**
- forward  
     differences, **297**  
     solution, 58  
     substitution, **58**
- frontal method, **66**
- FSEP, *see* feasible, solution to the equality problem
- full correction, **681**
- function  
     barrier, *see* barrier, function  
     element, **360**  
     forcing, **53**, 250, 511, 512, 514, 521, 524, 539  
     Huber, 770  
     merit, *see* merit function  
     nonsmooth, **33**  
     objective, *see* objective function  
     partially separable, **360**, 361–376  
     penalty, *see* penalty, function  
     separable, **359**
- Gelfand triple, **279**
- generalized  
     finite differences, 328–331  
     inverse, **20**, 181
- generalized Lanczos trust-region method, *see* algorithm
- Gershgorin bounds, 19
- gradient, **27**  
     error, 280  
     relative, **280**  
     finite differences, **297**, 301  
     generalized, **33**  
     in infinite dimensions, **274**  
     reduced, **41**
- gradient-related, **268**, 380–387
- hard case, **181**, 224  
     for eigenvalue-based approaches, 234  
     for Lanczos-based approaches, 227  
     with the absolute-value trust-region, 239
- Harwell Subroutine Library, *see* software
- heapsort, 791
- Hessian matrix, **27**  
     finite differences, **299**, 301  
     in infinite dimensions, **275**  
     reduced, **43**, 629
- Hilbert space, **275**
- Hölder inequality, **21**
- hook step, **175**
- Huber function, 770
- hyperplane, **26**
- ill-conditioned, 53, **61**
- inactive constraint, **39**, 755
- inequality-constrained quadratic programming (IQP), **631**
- inertia, **17**
- infinite dimensions, **274**
- initial trust-region radius, *see* radius, initial
- initial values for interval of uncertainty, 192
- inner product  
     in infinite dimensions, **275**  
     on  $\mathbb{R}^n$ , **16**, 21
- interior convergence, **189**
- interlacing property, **19**
- internal doubling, 161, 394, **394**, 788  
     algorithm, **394**, 394
- interpolation  
     blocks, **324**  
     determinant, **323**  
     multivariate, 324  
     path, **330**  
     pivot, **324**  
     polynomials, **324**  
         Lagrange fundamental, **324**, 336  
         Newton fundamental, **324**, 326–336
- set, **322**, 323–336  
     poised, **323**, 324

- to adjust radius, 783
- interval of uncertainty, **188**, 189
  - updating, **188**
- invariant subspace, **102**
- IQP, *see* inequality-constrained quadratic programming
- irreducible tridiagonal matrix, **224**
- isolated
  - limit point, 157, 258
  - minimizer, **37**, 143, 146, 157, 354
- iterate, **6**, 115
- iteration
  - successful, **117**
  - unsuccessful, **117**
  - very successful, **117**
- iterative refinement, **111**
- Jacobian matrix, **27**
- Karush–Kuhn–Tucker (KKT)
  - conditions, **41**
  - point, **41**
- Kronecker delta, **15**
- Krylov
  - space, **81**
  - degenerate, **92**
  - orthonormal basis for, 92
  - subspace, **81**
- Kullback–Leibler distance, 120
- L-BFGS-B, 460
- $\ell_2$ -norm problem
  - characterization of solution, 171
  - convex case, 177
  - hard case, 180
  - nonconvex case, 179
  - scaled, 200
  - solution of, 176
    - with affine constraints, 207, 230
- $\ell_\infty$ -norm problem, 243
- Lagrange
  - function, *see* Lagrangian
  - multiplier, **40**, 461, 519
    - consistent, **474**
    - continuous least-squares estimate, 617
    - first-order estimate, **577**
      - for inequality constraint, 609
    - least-squares estimate, **474**, 577, 626
      - for inequality constraint, 609
  - Lagrangian, **41**, 539
    - augmented, *see* augmented Lagrangian
    - barrier, **497**
  - LANCELOT, *see* software, 794
  - Lanczos method, 91, 235
  - algorithm, **95**
    - preconditioned, **104**
      - preliminary, **103**
      - transformed, **103**
    - preconditioned, 102
    - relationship to conjugate gradient method, 95
    - truncated approach, 221
  - LAPACK, 20, 64
  - LCP, *see* linear complementarity problem
  - $LDL^T$  factorization, **63**
  - least-squares
    - Lagrange multiplier estimate, *see* Lagrange linear, 68
      - iterative methods for, 106
      - problem, **68**
    - nonlinear, 120, 749
      - problem, **749**
    - total, 759
  - Lebesgue measure, 276–277
  - linear
    - complementarity problem (LCP), **771**
    - dependence, **18**
    - independence, **18**
  - linesearch, 120, 336, 359, 376, 381, 399, 566, 776
    - Armijo, 128, 150, 376, 459, 627
    - Goldstein, 453, 459
  - LINPACK, 20
    - method, **191**, 200
  - Lipschitz
    - constant, **28**, 305
    - continuity, **28**, 151, 153, 255, 296, 304
      - local, **28**, 407, 772
  - log-barrier, *see* barrier
  - LOQO, 636
  - $LQ$  factorization, 59, **69**
  - $M$ -norm problem, 200
  - machine precision, 298, 299, 793
  - magical step, **387**, 387, 393
  - manifold, **76**
  - Maratos effect, **642**, 679, 766
  - mathematical programming with equilibrium constraints (MPEC), 422, 777
  - MATLAB Optimization Toolbox, 566
  - matrix
    - absolute value, **17**
    - banded, **59**
    - dense, **56**
    - diagonal, **58**
    - element, **66**
    - Givens, **59**
    - Householder, **59**

- identity, **16**
- indefinite, **17**
- modification, **67**
- orthonormal, **58**
- permutation, **58**
- plane-reflector, **59**
- plane-rotation, **59**
- positive definite, **17**
- positive semidefinite, **17**
- sparse, **56**
- square root, **17**
- structured, **56**
- triangular
  - lower, **58**
  - unit, **58**
  - upper, **58**
- tridiagonal, **59**
- maximum likelihood, 758
- measure
  - criticality
    - first-order, **250**, 255, 259, 268, 273, 382, 449, 452, 458, 510, 552, 558
    - general, **249**, 265, 391, 393
    - second-order, **271**
  - Lebesgue, 276–277
- merit function, 627, **636**, 636
- min-max problems, **761**
- MINEW, 220
- minimization of a quadratic function
  - in a conjugate subspace, 77
  - in a subspace, **76**
- minimizer
  - global, **37**, 753
  - local, **37**, 115
    - isolated, **37**
    - strict, **37**
  - model, *see* model, minimizer
- minimum-norm solution, **20**, 70
- MINOP, 220
- MINPACK, 758
- model, 1, **115**
  - composite, **302**
  - conditional, *see* model, valid
  - conic, **162**, 285
  - convex, **139**, 172, **177**
    - asymptotically, 141
  - curvature, 124
  - Gauss–Newton, **751**
  - interpolation, 322–336
  - minimizer, **131**, 169, **176**
    - algorithm to find, *see* algorithm, to find model minimizer
- algorithm to find, using eigensolution, **199**
- approximating, 201
- characterization in  $\ell_2$  norm, 174, 175
- conjugate gradient-based approaches, **202**
- eigenvalue-based approaches for, **231**
- in ideal norms, **236**
- $\ell_\infty$  norm, **243**
- interior solution, 172, 178, 186, 187, 192
- Lanczos-based approaches for, **221**
- lower bound, 190
- projection-based approaches for, **235**
- termination of algorithm to find, **194**, **196**
- nonconvex, 172
- nonquadratic, **745**
- order, **117**
- partially separable, **362**
- quadratic, 323, 384–386
- reduction
  - normal step, 696
  - tangential step, 665
- sufficient reduction, 3, 116, **123**, **148**, 262, 267, 270, 413, 457, 507, 510, 638, 663, 665, 696, 724
- tensor, 162, 285
- valid, **307**, **308**, 334–335, 342, 406
  - second-order, **319**, 321, 342
  - with memory, **336**
- Moore–Penrose generalized inverse, **20**, 181
- Moreau decomposition, **36**, 449, 467, 470
- MPEC, *see* mathematical programming with equilibrium constraints
- multifrontal method, **66**
- NCP, *see* nonlinear complementarity problem
- negative curvature, *see* curvature, negative
- network, 459
- neural networks, 11
- Newton
  - correction, **52**
  - equations, **52**, 586
  - step, **52**, 588, 607
  - system, **52**
- Newton’s method, *see* algorithm, Newton’s method
- NITRO, 710, 720
- NL2SOL, 285, 758
- noise, **399**
- nondifferentiable penalty function, **610**
- nonlinear
  - complementarity problem (NCP), **770**

- function, **771**
- equations, *see* equations, nonlinear
  - in other norms, **759**
  - in the  $\ell_2$  norm, *see* equations, nonlinear
  - ear
  - fitting, **749**, **759**
  - least squares, *see* least squares, nonlinear
  - ear
- nonsmooth
  - problem, **47**, **407**
    - algorithms for, **408**, **423**
    - computing generalized gradients, **426**
    - models of, **434**
- norm, **20**
  - basic properties, **20**
  - bidual, **260**, **527**
  - dual, **259**, **531**
  - equivalence
    - one-sided, **252**
    - uniform, **123**, **166**, **363**, **460**, **531**
    - uniform, in set, **253**
    - weak, **272**, **273**
  - matrix, **22**
    - consistent, **23**
    - Euclidean, **22**
    - Frobenius, **22**
    - induced, **23**
    - max, **22**
    - submultiplicative property, **23**
    - subordinate, **23**
    - sum, **22**
  - vector, **21**
    - $B$ , **22**
    - $H$ , **22**
    - $\ell_p$ , **21**
    - dual, **21**
    - Euclidean, **16**, **21**
- normal
  - cone, *see* cone, normal
  - equations, **68**, **70**, **71**
  - step, *see* step, normal
  - model, **695**
- NP, **171**
  - complete, **171**
  - hard, **171**
- NS02, **758**
- null-space
  - approach, **72**, **110**
  - basis, **71**
- objective function, **1**, **37**, **115**
  - partially separable, **360**, **361–376**, **758**
- optimal control, *see* control, optimal, **609**, **720**
- optimality conditions, **37**
  - constrained, **39**
    - first-order necessary, **41**
    - second-order necessary, **42**
    - second-order sufficient, **42**
    - strong second-order sufficient, **43**
    - weak second-order necessary, **43**, **488**
  - convex, **44**
  - nonsmooth, **47**
    - first-order necessary, **48**
    - second-order sufficient, **49**
    - with constraints, **49**
  - unconstrained, **38**
    - first-order necessary, **38**
    - in infinite dimensions, **279**
    - second-order necessary, **38**
    - second-order sufficient, **39**
- order-simplex, **444**
- ORDPACK, **120**
- $p$ -integrable functions, **277**
- partial correction, **681**, **692**
- partially separable function, **120**, **282**, **360**, **361–376**
- partitioned updating, *see* quasi-Newton, partitioned updating
- path
  - conjugate gradient, *see* conjugate gradient path
  - dogleg, **219**
  - double-dogleg, **219**
  - interpolation, **330**
  - projected-gradient, **247**, **444**, **444**, **453**, **459**, **489**
- penalty
  - function, **574**
    - $\ell_1$ , **610**
    - $\ell_2$ , **660**
    - $\ell_\infty$ , **610**
    - exact, **613**
    - nonsmooth, **610**
    - quadratic, **575**, **582**
    - smooth, exact, **616**, **619**
  - parameter, **575**, **583**
- perturbation
  - bound, **61**
  - theory for mathematical programs, **47**
- pivot, **64**
  - 1 by 1, **65**
  - 2 by 2, **65**
    - interpolation, **324**
- poisedness, **323**, **324**, **330**
- polar, **35**, **449**
- PORT 3, **120**

- positive curvature, *see* curvature, positive  
 Powell-symmetric-Broyden update, 9  
 preconditioned Lanczos method, 102  
 preconditioner, 86, 87, 164
  - approximate inverse, 90
  - band, 90
  - diagonal, 90
  - element by element, 90
  - full, 90
  - incomplete Cholesky factorization, 90
 preconditioning, 86  
 primal barrier algorithm, *see* algorithm, barrier, primal  
 primal-dual barrier algorithm, *see* algorithm, barrier, primal-dual  
 problem
  - constrained, 37
  - structure, 359
  - unconstrained, 37
  - with equilibrium constraints, 422, 777
 projected-gradient path, 444, 444  
 projection
  - onto a cylinder, 443
  - onto an ordered simplex, 444
  - onto a set, 36, 441
    - inexact, 489
    - onto a sphere, 443
    - onto a subspace, 71
 proximal point method, 120  
 QCQP, *see* quadratic program, quadratically constrained  
 quadratic program
  - convex, 44
  - equality constrained, 72
  - general, 634
  - methods for solving, 633
  - optimality conditions, 43
  - quadratically constrained (QCQP), 746
  - sequential, *see* sequential quadratic programming
 quasi-Newton, 280, 281, 282, 284, 296, 297, 301, 627, 629, 758, 788  
 equation, 281  
 limited-memory, 220, 282, 296  
 partitioned updating, 282, 296, 459, 627, 633, 758  
 sizing, 285  
 Rademacher's theorem, 29, 411, 773  
 radius, 6, 116, 284
  - initial, 133, 784
  - algorithm to choose, 787
 management, 117, 158, 284, 394, 653, 782  
 range-space
  - approach, 71, 110
  - basis, 71
 rank, 18
  - deficient, 18
  - full, 18
 rate of convergence, *see* convergence, rate  
 Rayleigh quotient, 19, 82, 272
  - inequality, 19
 Rayleigh-Ritz procedure, 95, 101  
 reciprocal barrier function, 517  
 recursive quadratic programming, *see* sequential quadratic programming  
 reduced
  - gradient, 41
  - Hessian, 43, 629
  - Hessian method, 630
 reducible
  - tridiagonal matrix, 224
 regression, *see* least-squares
  - orthogonal distance, 758
 regular, 34, 407
  - generalized complementarity problem, 775
 relative
  - boundary, 26
  - error in gradient, 280
  - interior, 26
 residual, 749  
 restoration procedure, 725  
 Ritz
  - pair, 101
  - value, 101
  - vector, 101
 robust fitting, 758  
 Rosenbrock's function, 118  
 RQP, *see* sequential quadratic programming  
 saddle point, 39, 143, 148, 149  
 safeguarding parameters, 189
  - algorithm for updating, 189
 scaling, 117, 162, 163–166, 168, 279, 527, 543, 555, 566, 567, 569, 759
  - affine, 550
 Schur complement, 63  
 secant equations, 281  
 second-order correction, 643, 679, 691  
 secular equation, 182, 199
  - Newton's method for, 182
  - safeguarding, 185
 self-concordant barrier, 497  
 semibandwidth, 59  
 semidefinite programming, 498

- separable function, **359**  
 separating hyperplane theorem, **31**  
 sequential linear programming (SLP)  
     method, 745  
 sequential quadratic programming (SQP)  
     asymptotic convergence  
         equality problems, 624  
         inequality problems, 631  
     augmented Lagrangian methods, 637  
     Byrd–Omojokun-like approaches, 694  
         asymptotic convergence of, 705  
         convergence to second-order critical points, 708  
         global convergence of, 698  
     Celis–Dennis–Tapia-like approaches, 711  
     comparison of, 746  
     equality, **631**  
     filter method, 359, 721  
     for inequality constraints, 717  
     inequality, **631**  
     method, 623, **627**  
     nonsmooth exact penalty methods, 637  
         global convergence of, 638  
         problems with asymptotic convergence  
             of, 639  
         second-order corrections for, 642  
      $S\ell_1$ QP method, 639  
      $S\ell_\infty$ QP method, 639  
     smooth exact penalty methods, 655  
     Vardi-like approaches, 658  
         asymptotic convergence of, 679  
         convergence to second-order critical points, 690  
         global convergence of, 667  
 sequential unconstrained minimization technique  
     (SUMT), 542  
 set  
     active, **39**, 446  
     identification, 460  
     maximal, **468**  
     bounded, **25**  
     closed, **25**  
     closure, **26**  
     compact, **26**  
     convex, **30**, 450  
     interior, **26**  
     interpolation, *see* interpolation, set  
     open, **25**  
     relative boundary, **26**  
     relative interior, **26**  
     working, **245**  
 Sherman–Morrison–Woodbury formula, **57**  
 similar, **16**  
 similarity transform, **16**  
 simplicial decomposition, 460  
 singular  
     value decomposition, **17**  
     values, **18**  
     vectors, **18**  
 $S\ell_1$ QP method, **639**  
 $S\ell_\infty$ QP method, **639**  
 SLP, *see* sequential linear programming  
 Sobolev space, 279  
 software, 799  
     BECTR, 720  
     BOX-QUACAN, 459  
     BT, 423  
     CPLEX, 636  
     EISPACK, 20  
     ETR, 710  
     filterSQP, 744  
     GQTPAR, 200  
     Harwell Subroutine Library, 65, 66, 120,  
         200, 230, 243, 459, 758  
     HieLoW, 120  
     HSL\_VF05, 230  
     HSL\_VF06, 243  
     L-BFGS-B, 460  
     LANCZOS, 120, 285, 347, 394, 459, 609,  
         783, 788, 792, 794  
     LAPACK, 20, 64  
     LINPACK, 20  
     LOQO, 636  
     Matlab Optimization Toolbox, 566  
     MINEW, 220  
     MINOP, 220  
     MINPACK, 200, 758  
     NITRO, 710, 720  
     NL2SOL, 285, 758  
     NS02, 758  
     ORDPACK, 120  
     PATH, 777  
     PORT 3, 120  
     STENMIN, 120  
     TENSOLVE, 758  
     TRICE, 720  
     TRIDI, 200  
     TRON, 120, 459  
     UNCMIN, 120, 200, 285  
     VA06, 120  
     VA07, 758  
     VA21, 200  
     VB01, 758  
     VE08, 120, 285, 386, 459  
     VE09, 635  
     VE10, 285, 386, 459, 758

- VE12, 636  
 space  
     Banach, 274, 277  
     Hilbert, 275  
     spacer step, 387  
     sparse  
         matrix, 56  
         system, 61  
     sparsity, 56, 302  
     SPARSPAK, 64  
     spectral decomposition, 17  
     spectrum, 16, 276  
     SQP, *see* sequential quadratic programming  
     SR1 update, *see* symmetric rank 1 update  
     stability, 60  
     stabilization, 8  
     stationary point, 38  
     statistics, 10  
     steepest descent, 123, 221, 268, 277, 456, 784  
     Steihaug–Toint method, *see* algorithm, Steihaug–  
         Toint  
     STENMIN, 120  
     step, 2  
         affine, 567  
         composite, 658, 658, 694, 711, 721  
         horizontal, *see* step, tangential  
         magical, 387  
         normal, 658, 658, 662, 695, 711, 721  
         quasi-normal, *see* step, normal  
         spacer, 387  
         tangential, 658, 658, 662, 695, 721  
         transversal, *see* step, normal  
         trial, 116  
         vertical, *see* step, normal  
     stopping conditions, 794  
     stopping test, 117  
     structure  
         matrices, 57  
         problem, 359  
     subdifferential, 33  
     subgradient, 33  
     subproblem  
         approximate minimization, 131, 201  
         compatible, 724  
         exact minimization, 131  
         scaled, 164  
     subsequence, consistently active, 513, 514–  
         517, 523, 524, 532, 533, 540  
     subspace  
         affine, 542  
         invariant, 102  
         of element function, 360  
     Krylov, 81  
     projection onto a, 71  
     range, 360  
     SUMT, *see* sequential unconstrained minimiza-  
         tion technique  
     supporting hyperplane, 32  
     SVD, *see* singular, value decomposition  
     symmetric rank 1 update, 282, 283, 295, 296  
     tangent  
         cone, *see* cone, tangent  
     tangential  
         step, *see* step, tangential  
         model, 664, 695  
     Taylor approximation  
         first-order, 30  
         second-order, 30  
     Taylor’s theorem, 27  
     traffic assignment, 777  
     TRAM, 568  
     trial  
         point, 2, 116  
         step, 116  
     triangle inequality, 20  
     TRICE, 720  
     tridiagonal subproblem, 197, 222  
     TRON, 120, 459  
     trust region, 1, 115  
         applications, 9  
         element, 361  
         history, 8  
          $\ell_1$  norm, 170  
          $\ell_2$  norm, 170, 171  
          $\ell_\infty$  norm, 170, 243  
         radius, *see* radius  
         shape, 117, 119, 123, 162, 168, 261, 362,  
             363, 529, 551  
         structured, 359, 362  
         terminology, 9  
     trust-bundle method, 422  
     UNCMIN, 120, 285  
     unconstrained problem, 37, 38  
     VA06, 120  
     VA07, 758  
     Vardi-like approaches, *see* sequential quadratic  
         programming  
     variable, 1  
         dual, 43, 519–521, 536, 538  
         elemental, 360  
         selection, 769  
         slack, 548, 549, 719, 759  
     variational inequalities, 777  
     VB01, 758

- VE08, 120, 285, 386, 459  
VE09, 635  
VE10, 285, 386, 459, 758  
VE12, 636  
  
warm starts, 636  
watchdog technique, **350**, 358, 549, 655  
well-conditioned, **61**  
working set method, **244**, **245**

# Author Index

- Aazhang, B., 11  
Aboudheir, A., 11  
Adler, I., 636  
Agren, H., 11  
Alexandrov, N. M., 10, 162, 376  
Alizadeh, F., 498  
Allen, D. M., 12  
Allgower, E. L., 279  
Almlöf, J., 11  
Amaya, J., 220  
Andersen, E. D., 496  
Anderson, E., 20, 64  
Anderson, H., 11  
Andrews, D. A., 10, 175  
Anstreicher, K., 746  
Antreich, K., 11  
Ariyawansa, K. A., 162  
Ashcraft, C., 65  
Attia, N. F., 592  
Auer, G., 784  
Axelsson, O., 91
- Bai, Z., 20, 64  
Bakr, M. H., 11, 770  
Balakrishnan, V., 498  
Bandler, J. W., 11, 770  
Bannert, T., 426  
Barakat, R., 10  
Barlow, J. L., 248  
Bartels, R. H., 635  
Bartholomew-Biggs, M. C., 608, 637  
Bazaraa, M. S., 616  
Beale, E. M. L., 629  
Belegundu, A. D., 11  
Bell, B. M., 438  
Bellare, M., 248  
Ben-Tal, A., 11, 176, 235, 497, 498, 716  
Bereaux, Y., 12  
Bertsekas, D. P., 248, 460, 582, 608, 609  
Best, M. J., 444, 635  
Biegler, L. T., 11, 630, 633, 710  
Bielschowsky, R. H., 710  
Biemacki, R. M., 11, 770  
Bierlaire, M., 11, 12, 120, 473, 758
- Biggs, M. C., 637  
Billups, S. C., 10, 777  
Bischof, C., 20, 64  
Björck, Å., 59, 70, 71, 73, 107  
Bock, H. G., 10, 406  
Bockmann, C., 10  
Bofill, J. M., 11  
Boggs, P. T., 120, 620, 629, 636, 657, 748, 758, 759  
Böhmer, K., 279  
Bongartz, I., 343, 358  
Bonnans, J. F., 121, 359, 554, 629, 633  
Booker, A. J., 10  
Boot, J. C., 635  
Borggaard, J., 10, 11  
Borwein, J. M., 43  
Bouaricha, A., 120, 162, 207, 758, 759  
Bouhtou, M., 554  
Bowers, K. L., 10, 758  
Boyd, S., 498  
Boyle, J. M., 20  
Branch, M. A., 207, 221, 489, 566, 567  
Bregman, L. M., 120  
Breitfeld, M. G., 497  
Brimberg, J., 760  
Brown, L. D., 10, 280  
Broyden, C. G., 281, 296, 592, 630  
Buckley, A. G., 296  
Budil, D. E., 12  
Bulsari, A. B., 11  
Bulteau, J. P., 207, 220, 221  
Bunch, D. S., 758  
Bunch, J. R., 20, 65, 105, 635  
Burke, J. V., 121, 271, 285, 359, 451, 461, 473, 616, 716  
Burns, J., 10, 11  
Butcher, J. C., 10  
Byrd, R. H., 73, 120, 161, 207, 221, 272, 295, 380, 396, 460, 510, 549, 630, 678, 679, 690, 691, 694, 698, 710, 719, 720, 758, 759
- Calamai, P. H., 451, 460, 473  
Calzolari, G., 10

- Carpenter, T. J., 549, 636  
 Carroll, C. W., 517  
 Carter, R. G., 162, 271, 296, 297, 406  
 Case, L., 10  
 Cauchy, A., 139  
 Celis, M. R., 716  
 Cesari, A., 11  
 Chabriac, Y., 44  
 Chakravarti, N., 444  
 Chamberlain, R. M., 358, 654  
 Chan, T. F., 200  
 Chandra, R., 105  
 Chang, Y. Y., 635  
 Charalambous, C., 616, 658, 770  
 Chen, B., 776  
 Chen, C., 777  
 Chen, L., 282, 297  
 Chen, P. K., 162  
 Chen, S. H., 11, 770  
 Chen, X., 746, 776  
 Chen, X. D., 716  
 Chen, Z., 473, 716  
 Chen, Z. W., 721  
 Cheng, S., 68  
 Cheng, S. H., 68, 242  
 Cheng, Y. B., 11, 280  
 Chin, C. M., 745  
 Chow, T. T., 162  
 Chu, E. K. W., 359  
 Chvátal, V., 516, 635  
 Clarke, F. H., 34, 50  
 Clermont, J. R., 10, 12  
 Cline, A. K., 200  
 Coleman, T. F., 10, 73, 111, 207, 221, 248,  
     271, 274, 279, 301, 302, 394, 496,  
     554, 563, 565–569, 592, 616, 630,  
     633, 654, 678, 691, 694, 748, 758,  
     784  
 Collatz, L., 23  
 Colson, B., 422, 777  
 Concus, P., 91  
 Conn, A. R., 10, 11, 67, 68, 120, 147, 161,  
     200, 248, 271, 285, 295–297, 313,  
     336, 343, 347, 358, 376, 393, 406,  
     437, 450, 451, 459–461, 473, 489,  
     497, 542, 549, 550, 592, 608, 609,  
     616, 630, 633, 636, 654, 658, 691,  
     694, 770, 784, 788, 792, 794  
 Contesse, L. B., 43  
 Contreras, M., 285  
 Cooley, D. W., 200  
 Corliss, G., 302  
 Corradi, G., 162  
 Coster, J. E., 11  
 Cottle, R. W., 635  
 Coulman, P. K., 11, 770  
 Courant, R., 582  
 Crouzeix, J. P., 44  
 Cryer, C. W., 23, 27  
 Cullum, J., 105  
 Curtis, A. R., 301  
 Dafermos, S., 777  
 Daniel, J. W., 91, 210  
 Dantzig, G. B., 635  
 Das, I., 720  
 Davidon, W. C., 296  
 De Boor, C., 336  
 De Luca, T., 776  
 De Moor, B., 776  
 de Sampaio, R. J. B., 422, 758  
 De Schutter, B., 776  
 Dean, E. J., 10  
 Delalande, M. E., 10  
 Dem'yanov, V. F., 770  
 Dembo, R. S., 207, 248, 633  
 Demmel, J., 20, 64  
 Deng, N., 282, 297, 359  
 Dennis, J. E., 9, 10, 30, 51, 53, 120, 128, 162,  
     168, 200, 220, 282, 285, 301, 376,  
     396, 399, 421, 422, 437, 567, 711,  
     716, 720, 745, 748, 758, 759, 784,  
     798  
 Deuflhard, P., 279  
 Di, S., 162  
 Di Pillo, G., 620, 621  
 Diao, B., 655  
 Dikin, I. I., 550, 554  
 Diniz-Ehrhardt, M. A., 248, 459, 784  
 Dirkse, S. P., 777  
 Djang, A., 635  
 Domich, P. D., 636  
 Donaldson, J. R., 120  
 Dongarra, J. J., 20, 64  
 Dostál, Z., 248  
 Druskin, V., 106  
 Du, L., 220  
 DuCroz, J., 20, 64  
 Duff, I. S., 63–66, 437, 770  
 Dunn, J. C., 460, 461  
 Dussault, J. P., 434  
 Echebest, N., 200, 396  
 Ecker, J. G., 716  
 Eckmueller, J., 11  
 Eckstein, J., 120  
 Edlund, O., 10, 759

- Edsberg, L., 10  
 Einarsson, H., 423  
 Eisenstat, S. C., 207, 633  
 Ekblom, H., 10, 759  
 El-Alem, M., 399, 710, 711, 716, 745, 748, 759  
 El-Bakry, A. S., 517  
 El-Ghaoui, L., 498  
 El-Hallabi, M., 271, 386, 396, 399, 437, 745,  
     759  
 Elster, C., 406  
 Erisman, A. M., 63, 64  
 Eskow, E., 68, 242  
 Evans, D. J., 91  
 Facchinei, F., 359, 473, 621, 776  
 Fan, Y., 678, 721  
 Fei, X., 423  
 Felgenhauer, U., 296  
 Feng, G., 460  
 Fenyes, P. A., 630  
 Ferland, J. A., 434  
 Feron, E., 498  
 Ferris, M. C., 746, 777  
 Fiacco, A. V., 37, 43, 44, 47, 495, 517, 542,  
     582, 592  
 Fichter, U., 11  
 Fischer, A., 776  
 Fletcher, R., 9, 34, 37, 43, 50, 65, 73, 91, 161,  
     248, 281, 296, 394, 422, 426, 433,  
     434, 437, 616, 620, 633, 635, 654,  
     744–746, 758  
 Flippo, O. E., 176  
 Fontecilla, R., 630, 633  
 Fortin, M., 609  
 Fourer, R., 496  
 Fox, P. A., 120  
 Fraley, C., 799  
 Frangioni, A., 10, 423  
 Frank, M., 635  
 Frank, P. D., 10  
 Freed, J. H., 12  
 Freund, R. M., 497  
 Friedlander, A., 248, 459, 461, 549  
 Frisch, K. R., 495  
 Fromme, S., 12  
 Fromovitz, S., 633  
 Fu, M., 716  
 Fuentes, O., 11  
 Fugger, P., 235  
 Fujisawa, K., 498  
 Fukushima, M., 10, 279, 422, 426, 616, 654,  
     758, 776  
 Furey, B. P., 11  
 Gabriel, S. A., 438, 777  
 Gallo, G., 10, 423  
 Gander, W., 200  
 Gao, L., 10, 770  
 Gao, Z., 655  
 Garbow, B. S., 20, 758  
 Garey, M. R., 170  
 Garfield, E., 8  
 Gay, D. M., 11, 120, 161, 168, 175, 199, 200,  
     285, 549, 758, 798  
 George, A., 63, 64  
 Germain, M., 12  
 Gertz, E. M., 386, 399  
 Gilbert, J. Ch., 121, 549, 630, 633, 679, 710,  
     719  
 Gill, P. E., 9, 37, 43, 68, 73, 91, 242, 248, 301,  
     434, 495, 496, 608, 635, 636, 798  
 Glad, T., 620, 621  
 Glowinski, R., 609  
 Goemans, M. X., 498  
 Goldfarb, D., 281, 296, 301, 635, 636, 746  
 Goldfeldt, S. M., 8, 175, 199  
 Goldstein, A. A., 459  
 Golub, G. H., 20, 59, 61, 64, 71, 91, 92, 105,  
     111, 200, 635  
 Gomes, F. A. M., 710, 720  
 Gomes-Ruggiero, M. A., 248, 459, 784  
 Gondzio, J., 496  
 Gonzaga, C. C., 554  
 Gonzalez-Lima, M. D., 549  
 Goode, J. J., 616  
 Gopal, V., 11  
 Gosselin, O., 10  
 Gould, F. J., 43  
 Gould, N. I. M., 44, 67, 68, 111, 120, 147,  
     161, 200, 230, 242, 243, 248, 271,  
     285, 295–297, 343, 344, 347, 358,  
     376, 380, 406, 450, 451, 459–461,  
     473, 489, 495, 497, 517, 542, 549,  
     550, 592, 593, 608, 609, 630, 633,  
     635, 636, 745, 784, 788, 792, 794  
 Gow, A. S., 11  
 Grace, A., 566  
 Graeb, H., 11  
 Greenbaum, A., 20, 64, 91, 106  
 Greenstadt, J., 68, 242  
 Griewank, A., 296, 302, 633  
 Griffith, R. E., 8  
 Grimes, R. G., 65  
 Grippo, L., 359, 620, 621  
 Grothey, A., 423  
 Gruver, W. A., 30  
 Guardarucci, M. T., 200, 396

- Guo, X. Z., 10, 11  
 Gurwitz, C., 630
- Haberland, D., 11  
 Hackbusch, W., 91  
 Haddou, M., 422  
 Hager, W. W., 230, 608, 633  
 Hall, A. D., 120  
 Hammarling, S., 20, 64  
 Han, C., 248, 554  
 Han, J., 10, 162, 359, 567, 716  
 Han, J. Y., 721  
 Han, Q., 162  
 Han, S. P., 616, 621, 654, 716  
 Hanke, M., 10, 758  
 Hanson, R. J., 59, 71, 73, 162, 359, 758  
 Haring, R. A., 11, 770  
 Harker, P. T., 776  
 Hasegawa, T., 616  
 He, G., 655  
 Hebden, M. D., 9, 199  
 Heinkenschloss, M., 207, 221, 279, 399, 567,  
     716, 720, 784  
 Heinz, J., 655  
 Helfrich, H. P., 10, 759  
 Helgaker, T., 11  
 Hempel, C., 592  
 Hestenes, M. R., 91, 582, 608  
 Higham, D. J., 162  
 Higham, N. J., 8, 23, 59, 61, 64, 65, 68, 71,  
     111, 200, 242  
 Hillstrom, K. E., 758  
 Hiriart Urruty, J. B., 422  
 Holstad, A., 11  
 Holt, J. N., 758  
 Horaud, R., 11  
 Hozack, R. S., 11  
 Hribar, M. E., 111, 549, 710, 720  
 Huanchen, W., 423  
 Huang, L. R., 50, 616  
 Hulbert, L. A., 248
- Ibaraki, T., 422, 426, 616  
 Idnani, A., 635  
 Ikebe, Y., 20  
 Irons, B. M., 66  
 Iusem, A. N., 120
- Jackiewicz, Z., 10  
 Jackson, M. P., 248  
 Jagersand, M., 11  
 Jain, V. K., 11  
 Jalali, J., 11  
 Jansen, B., 176
- Jarausch, H., 759  
 Jarre, F., 396, 497, 746  
 Jensen, D., 497  
 Jensen, H. J. A., 11  
 Jensen, J. J. A., 11  
 Ji, X. S., 11  
 Jiang, H., 438, 776  
 Jittorntrum, K., 496  
 Johnson, D. S., 170  
 Jonsson, O., 11, 121  
 Jorgensen, P., 11  
 Júdice, J. J., 248, 473
- Kabadi, S. N., 43, 247, 489  
 Kamat, M. P., 11  
 Kaniel, S., 91, 106  
 Kantorovich, L., 53  
 Kanzow, Ch., 776  
 Karisch, S. E., 10  
 Karmarkar, N., 495  
 Karush, W., 43  
 Kaufman, L. C., 65, 220, 285, 635  
 Kawaguchi, H., 11  
 Ke, X., 359, 716  
 Kearsley, A. J., 620, 629  
 Kehtarnavaz, N., 12  
 Keller, E. L., 635  
 Kelley, C. T., 10, 279  
 Keyes, D. E., 10  
 Khalfan, H. F., 295  
 Kim, N. H., 437  
 Kiwi, K. C., 121, 422  
 Klema, V. C., 20  
 Knizhnerman, L., 106  
 Knoth, O., 161  
 Knuth, D. E., 51, 792  
 Kojima, F., 10, 11  
 Kojima, M., 498  
 Koontz, J. E., 200, 285  
 Kritpiphat, W., 11  
 Krogh, F. T., 162, 359, 758  
 Kruk, S., 716, 746  
 Kuhn, H. W., 43  
 Kwok, H. H., 11
- Lalee, M., 710, 711  
 Lampariello, F., 359  
 Lancaster, P., 23  
 Lanczos, C., 20, 105  
 Lannes, A., 10  
 Larsson, T., 11, 121  
 Lasdon, L. S., 437, 549, 721  
 Lau, D. T. M., 162  
 Laudise, R. A., 11

- Launay, G., 629, 633  
 Lawrence, C. L., 549  
 Lawson, C. L., 59, 71, 73  
 Layn, K. M., 11  
 Le Tallec, P., 609  
 Le Thi, H. A., 11, 175, 236, 716  
 Lee, S., 12  
 Leibfritz, F., 10  
 Lemaire, B., 434  
 Lemaréchal, C., 121, 358, 422, 654  
 Lescrenier, M., 67, 461, 473  
 Levenberg, K., 8, 121, 758, 759, 813  
 Levitin, E. S., 459  
 Lewis, J. G., 65  
 Lewis, R. M., 10, 11, 162  
 Leyffer, S., 744–746  
 Li, D., 777  
 Li, S. B. B., 421, 422, 437  
 Li, W., 248, 776  
 Li, Y., 207, 221, 248, 271, 274, 394, 496, 554,  
     563, 565–569, 769, 770, 784  
 Liang, X., 460  
 Liao, A., 10, 279  
 Lin, C., 91, 120, 459  
 Lipkin, H., 11  
 Liu, D., 296  
 Liu, D. L., 11  
 Liu, G., 10, 359  
 Liu, J., 248, 554  
 Liu, J. W. H., 63, 64  
 Liu, S. C., 636, 746  
 Liu, X., 710  
 Lootsma, F. A., 495, 535, 592, 608  
 Love, R. F., 760  
 Lu, P., 460  
 Lucia, A., 10, 11  
 Lucidi, S., 175, 200, 230, 344, 359, 380, 621  
 Luder, E., 758  
 Luenberger, D. G., 27, 32, 36, 47, 91, 592  
 Lukšan, L., 10, 230, 280, 758, 759  
 Luo, Z. Q., 716, 776  
 Lustig, I. J., 549, 636  
 Lüthi, H. J., 12  
 Lyle, S., 175  
 Mészáros, C., 496  
 Maany, Z. A., 746  
 Maciel, M. C., 710, 711, 716, 720, 748  
 Mackens, W., 759  
 Maculan, N., 10  
 Madsen, K., 9–11, 423, 437, 759, 770  
 Madyastha, R. K., 11  
 Mallick, M. K., 11, 759  
 Malozemov, V. N., 770  
 Maly, T., 11, 758  
 Mangasarian, O. L., 37, 43, 616, 621, 633, 776,  
     777  
 Maratos, N., 616, 642, 654  
 Marquardt, D., 8, 121, 175, 758, 759, 813  
 Martinet, B., 120  
 Martínez, J. M., 10, 175, 200, 207, 248, 396,  
     438, 459, 461, 549, 710, 716, 720,  
     758  
 Maurer, H., 279  
 Mauricio, D., 10  
 Mayne, D. Q., 616, 654, 655  
 McAfee, K. B., 11  
 McCartin, B. J., 220  
 McClellan, T. E., 11  
 McCormick, G. P., 37, 43, 459, 495, 517, 542,  
     582, 592  
 McKenna, M. P., 248  
 McKenney, A., 20, 64  
 McKinnon, K., 423  
 McMurray, G. V., 11  
 Mehrotra, S., 496, 716, 746  
 Mei, H. H. W., 9, 220, 396, 784  
 Mentel, J., 11  
 Mesirov, J. P., 248  
 Mifflin, R., 336, 495, 550  
 Mine, H., 616  
 Mittelmann, H. D., 10  
 Mjolsness, E., 11  
 Moler, C. B., 20, 200  
 Moller, E., 11  
 Mongeau, M., 616  
 Monteiro, R. D. C., 554, 636  
 Moré, J. J., 9, 91, 120, 147, 161, 168, 199,  
     200, 248, 271, 296, 301, 302, 451,  
     459–461, 473, 592, 716, 758, 777,  
     799  
 Moreau, J. J., 36  
 Moretti, A. C., 438  
 Morgan, A. P., 10  
 Morrill, G. L., 11, 770  
 Morrison, D. D., 8, 121, 175, 758, 759, 813  
 Mukai, H., 10, 621  
 Mukai, K., 10  
 Mullani, N., 12  
 Muller Liebler, G., 11  
 Mulvey, J. M., 549, 636  
 Munksgaard, N., 65  
 Murray, W., 9, 37, 43, 68, 73, 91, 242, 248,  
     301, 434, 495, 496, 535, 592, 608,  
     630, 632, 633, 635–637, 798  
 Murtagh, B. A., 635  
 Murty, K. G., 43, 247, 489

- Nabona, N., 220  
 Nagurney, A., 777  
 Nakata, K., 498  
 Nash, S. G., 106, 207, 296, 495, 496, 542, 550  
 Nazareth, J. L., 297  
 Neilsen, H. B., 65  
 Nelson, R., 11  
 Nelson, S. A., 376  
 Nemirovskii, A., 497, 498, 746  
 Nesterov, Y., 497  
 Neumaier, A., 406  
 Ng, K. F., 50, 616  
 Nguyen, H. v., 422  
 Niemi, R. D., 716  
 Noble, B., 210  
 Nocedal, J., 111, 242, 243, 296, 386, 437, 460,  
     549, 630, 633, 679, 710, 711, 719,  
     720, 770  
 Notay, Y., 91  
 O'Leary, D. P., 91, 248  
 Olkin, J. A., 200  
 Omojokun, E. O., 698, 710, 711  
 Orban, D., 161, 248, 495, 542, 549, 550, 636,  
     784  
 Ortega, J. M., 51, 53, 274  
 Osborne, M. R., 9, 120, 437, 496, 635, 758,  
     759, 769, 770  
 Ostrouchov, S., 20, 64  
 Outrata, J., 423  
 Overton, M. L., 549, 630  
 Paige, C. C., 105–107, 758  
 Palagi, L., 175  
 Panattoni, L., 10  
 Pang, J. S., 438, 635, 776, 777  
 Panier, E., 359, 655  
 Pantoja, J. F. A., 616, 655  
 Papadimitriou, C. H., 170  
 Papalambros, P. Y., 376  
 Pardalos, P., 248, 554  
 Park, K., 746  
 Parlett, B. N., 20, 65, 101, 105, 106, 230  
 Patriksson, M., 777  
 Pedersen, H. C., 358, 654  
 Peng, J., 716, 776  
 Perry, A., 296  
 Petzold, L. R., 11, 758  
 Pham Dinh, T., 10, 11, 175, 236  
 Phan huy Hao, E., 716, 746  
 Phong, T. Q., 11  
 Piepmeyer, J. A., 11  
 Pietrzykowski, T., 616  
 Pires, F. M., 248  
 Plantenga, T. D., 710, 711, 720  
 Plassman, P. E., 758  
 Plummer, J., 549  
 Pola, C., 554  
 Polak, E., 10, 608, 620, 621, 654, 770  
 Poliquin, R., 422  
 Poljack, S., 10  
 Polyak, B. T., 111, 248, 459, 460  
 Polyak, R., 496, 497  
 Ponceleón, D. B., 68, 242, 496  
 Ponnambalam, K., 799  
 Pornbacher, F., 11  
 Pothen, A., 73  
 Potra, F. A., 279, 549  
 Powell, M. J. D., 9, 86, 120, 139, 220, 285,  
     297, 301, 302, 336, 358, 399, 437,  
     497, 582, 608, 616, 629, 635, 654,  
     657, 678, 716, 759, 770, 788  
 Prieto, F. J., 633  
 Propoi, A. I., 10  
 Pschenichny, B. N., 629, 654  
 Psiaki, M., 746  
 Pukhlikov, A. V., 10  
 Qi, L., 422, 438, 776, 876  
 Quan, L., 11  
 Quandt, R. E., 8, 175, 199  
 Raydan, M., 248  
 Reaser, M. H., 11  
 Reid, J. K., 63–66, 91, 220, 230, 301, 437, 758,  
     759, 770  
 Reinsch, C., 199  
 Ren, Y. H., 11, 758  
 Renaud, J. E., 11  
 Rendl, F., 10, 235  
 Rheinboldt, W. C., 51, 53, 274, 279  
 Ritter, K., 635  
 Robinson, S. M., 44, 632, 776  
 Rockafellar, R. T., 27, 30, 32, 34, 36, 47, 120,  
     593, 608, 609  
 Rodríguez, J. F., 11  
 Rogaway, P., 248  
 Roger, A., 10  
 Rogers, J. E., 636  
 Rojas, M., 10, 235, 758  
 Roma, M., 175, 200, 230, 344, 380  
 Ron, A., 336  
 Rosenbrock, H. H., 118  
 Roy, R., 12  
 Rudnicki, M., 11, 758  
 Ruedenberg, K., 11  
 Saad, Y., 20, 91

- Sachs, E. W., 10, 12, 30, 279, 609  
 Sadjadi, S. J., 799  
 Sagara, N., 758  
 Sagastizábal, C. A., 121  
 Sahba, M., 616  
 Sainz de la Maza, E., 746  
 Salane, D. E., 758  
 Sandler, B. H., 10  
 Santos, S. A., 235, 248, 459, 461, 549, 716, 758, 784  
 Sargent, R. W. H., 549, 629  
 Sarkar, S., 721  
 Sarkar, T. K., 11  
 Sartenaer, A., 147, 271, 296, 376, 406, 451, 459–461, 473, 489, 542, 550, 609, 616, 784, 788  
 Sauer, Th., 331, 336  
 Saunders, M. A., 68, 73, 105, 107, 242, 301, 495–497, 608, 635, 636, 758  
 Saxen, H., 11  
 Saxena, S., 12  
 Schaepperle, J., 758  
 Scheinberg, K., 10, 313, 336, 746  
 Schittkowski, K., 608  
 Schleiff, S., 10  
 Schlick, T., 68  
 Schlöder, J. P., 10, 406  
 Schmid, C., 630, 633, 710  
 Schnabel, R. B., 9, 30, 51, 53, 68, 73, 120, 128, 161, 162, 168, 200, 207, 221, 242, 272, 282, 285, 295, 301, 380, 396, 678, 690, 691, 694, 758, 759, 784, 798  
 Schneur, R., 497  
 Scholtes, S., 10, 422, 777  
 Schramm, H., 422, 423  
 Schryer, N. L., 120  
 Schulz, V. H., 10, 406  
 Schulze, M., 12  
 Schwartz, G., 11  
 Schwenger, R., 11  
 Schwetlick, H., 758  
 Scolnik, H. D., 200, 396  
 Sebudandi, Ch., 10, 784  
 Semple, J., 10  
 Serafini, D. B., 10  
 Sevick Muraca, E. M., 12  
 Shahabuddin, J. S., 376, 784  
 Shanno, D. F., 281, 296, 497, 549, 636  
 Sheffi, Y., 777  
 Shiina, T., 11  
 Shultz, G. A., 120, 161, 207, 221, 272, 380, 396, 678, 690, 691, 694  
 Si, J., 11  
 Sillanpaa, M., 11  
 Simantiraki, E. M., 549  
 Sinclair, J. W., 437  
 Smale, S., 171  
 Smith, B. T., 20  
 Smith, R. C., 10, 758  
 Snyman, J. A., 11  
 Soares, J., 473, 776  
 Sofer, A., 207, 495, 496, 542, 550  
 Solodov, M. V., 776  
 Sorensen, D. C., 9, 20, 64, 73, 147, 161, 168, 175, 199, 200, 235, 592  
 Spellucci, P., 655  
 Stander, N., 11  
 Steiglitz, K., 170  
 Steihaug, T., 9, 207, 554, 630, 633  
 Stern, R. J., 176, 235, 716  
 Stewart, G. W., 20, 73, 200, 301  
 Stewart, R. A., 8  
 Stiefel, E., 91  
 Stoer, J., 91, 105  
 Stöhr, M., 10, 422, 777  
 Stone, R. E., 635  
 Strakoš, Z., 91  
 Strodiot, J. J., 422  
 Studer, G., 12  
 Sugimoto, T., 422  
 Sun, D., 776  
 Sun, J., 422, 437, 554, 636, 716, 746, 876  
 Sun, J. Q., 11  
 Sun, L. P., 207, 221  
 Sun, W., 162, 422, 758  
 Sunar, M., 11  
 Sunder, W. A., 11  
 Sutherland, W. A., 27  
 Swetits, J., 248  
 Sykulska, J. K., 11, 280  
 Szularz, M., 175  
 Tanabe, T., 720, 721  
 Tanaka, Y., 616  
 Tapia, R. A., 271, 285, 399, 421, 422, 437, 549, 603, 608, 630, 716, 745, 759  
 Tappayuthpijarn, C., 11  
 Tatsumi, K., 10  
 Teboulle, M., 120, 176, 235, 716  
 Ternet, D. J., 721  
 Terpolilli, P., 10, 438  
 Thomas, S., 139, 399  
 Tibshirani, R., 770  
 Tiller, V., 11, 758  
 Tinney, W. F., 64  
 Tismenetsky, M., 23

- Tits, A. L., 359, 549, 608, 655, 770  
 Todd, M. J., 497  
 Toint, Ph. L., 9–12, 67, 68, 120, 147, 161, 162,  
     200, 207, 230, 248, 271, 279, 285,  
     295–297, 301, 302, 313, 336, 343,  
     344, 347, 357–359, 376, 380, 386,  
     406, 450, 451, 459–461, 473, 489,  
     497, 517, 542, 549, 550, 554, 592,  
     608, 609, 630, 633, 636, 745, 746,  
     758, 784, 788, 792, 794  
 Tolle, J. W., 43, 248, 620, 629, 657, 748  
 Tomlin, J. A., 495  
 Tong, X., 459  
 Tontiwachwuthikul, P., 11  
 Toraldo, G., 248, 271, 451, 460, 461, 473  
 Torczon, V., 10, 162, 399, 784  
 Trosset, M. W., 10  
 Trotter, H. F., 8, 175, 199  
 Tse, E., 248, 554, 636  
 Tseng, P., 549, 776  
 Tsoutsias, D. I., 11  
 Tsuchiya, T., 554, 636  
 Tucker, A. W., 43  
 Tulowitzki, U., 248  
 Tuyttens, D., 473, 758  
 Ulbrich, M., 279, 359, 473, 567, 745, 777  
 Ulbrich, S., 279, 359, 473, 567, 745  
 Urban, T., 549  
 Vacchino, C., 200, 396  
 Van de Panne, C., 635  
 van der Vorst, H. A., 64  
 Van Huffel, S., 759  
 Van Loan, C. F., 20, 59, 61, 64, 71, 92, 105,  
     111, 200  
 Vandenberghe, L., 498  
 Vanderbei, R. J., 235, 636  
 Vandergraft, J. S., 10  
 Vandewalle, J., 759  
 Vardi, A., 678, 770  
 Vavasis, S. A., 170, 176, 247, 636  
 Vial, J. P., 207, 220, 221  
 Vicente, L. N., 10, 175, 297, 380, 393, 399,  
     567, 711, 720, 784  
 Viswesvariah, C., 11, 297, 393, 770  
 Vlček, J., 10, 280, 758, 759  
 Vogel, C. R., 10, 758  
 von Matt, U., 200  
 Walker, H. F., 207  
 Walker, J. W., 64  
 Wang, S., 10, 11, 746  
 Wang, X., 10, 11, 459  
 Wang, Y., 554  
 Watson, G. A., 50, 770  
 Watson, L. T., 10, 11  
 Wedin, P. A., 10  
 Weigmann, A., 121, 285, 359  
 Weihs, C., 10  
 Weiss, B. E., 200, 285  
 Welsch, R. E., 285, 758, 798  
 Whinston, A., 635  
 Wilkinson, J. H., 20, 61, 64, 111, 200  
 Williams, J. W. J., 792  
 Williamson, K. A., 10, 221, 716, 745, 759  
 Willoughby, R. A., 105  
 Wilson, R. B., 629  
 Win, M. Z., 12  
 Winfield, D., 9, 335  
 Wloka, J., 279  
 Wolfe, P., 635  
 Wolkowicz, H., 10, 176, 235, 297, 716, 746  
 Womersley, R. S., 34, 654  
 Wonnacott, R. J., 406  
 Wonnacott, T. H., 406  
 Wright, M. H., 9, 37, 43, 73, 91, 242, 301,  
     434, 495, 496, 549, 550, 608, 630,  
     632, 635, 636, 798  
 Wright, S. J., 43, 120, 422, 495, 496, 633, 758,  
     799  
 Xiao, B., 776  
 Xiao, Y., 220, 357–359  
 Xu, C., 220, 437, 460, 678  
 Xu, D., 359  
 Xu, J., 11  
 Xu, X., 496  
 Xu, Y., 331, 336  
 Yabe, H., 655, 720, 721  
 Yamakawa, E., 422, 426  
 Yamamoto, Y., 10, 279  
 Yamashita, H., 608, 655, 720, 721  
 Yamashita, N., 776  
 Yang, B., 655  
 Yang, E. K., 248  
 Yang, Y., 777  
 Yassine, A., 10, 11  
 Ye, Y., 176, 248, 554, 636, 716  
 Yin, H., 567  
 Yosida, K., 275  
 You, Z., 655  
 Yu, G., 549  
 Yuan, J., 422, 758  
 Yuan, W., 678, 748

Yuan, Y., 139, 161, 162, 218, 359, 386, 387,  
394, 399, 426, 437, 616, 654, 655,  
657, 710, 716, 746, 770

Zapf, H., 11

Zarantonello, E. H., 36, 450

Zavriev, S. K., 746

Zenios, S. A., 248

Zhang, J., 220, 282, 297, 437, 633, 678, 746

Zhang, K., 655

Zhang, X., 549

Zhang, Y., 549, 630, 716, 776

Zhao, M., 11

Zhou, F., 220, 359

Zhou, G., 11

Zhou, J., 770

Zhou, J. L., 359

Zhou, S., 777

Zhu, C., 121, 460

Zhu, D., 451, 459, 633, 655, 678

Zhu, M., 297

Zhu, T., 10, 280

Zibulevsky, M., 11, 497

Zippel, R., 176

Zorkaltsev, V. I., 554

Zowe, J., 279, 422, 423

Zupke, M., 776

Zwick, D., 10, 759